SISTEMA DE INFORMACION WEB PARA LA GESTION DE CLIENTES (CRM) PARA LA EMPRESA MI-MARTINEZ

Edward Hair Díaz Alba
Manuel Alejandro Pinzón Salgado



UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
FACULTADAD TECNOLOGICA
TECNOLOGIA EN SISTEMATIZACION DE DATOS
BOGOTA
2015

SISTEMA DE INFORMACION WEB PARA LA GESTION DE CLIENTES (CRM) PARA LA EMPRESA MI-MARTINEZ

Edward Hair Díaz Alba
Manuel Alejandro Pinzón Salgado

Trabajo de grado para optar por el título de “Tecnólogo en Sistematización de Datos”

TUTOR
Ingeniera Roció Rodríguez

UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
FACULTADAD TECNOLOGICA
TECNOLOGIA EN SISTEMATIZACION DE DATOS
BOGOTA
2014

Nota de aceptación

_____
_____
_____
_____
_____

_____
Jurado

_____
Jurado

Bogotá, Junio 1 del 2015

Este trabajo es dedicado a Dios por darnos la fortaleza y perseverancia de seguir adelante, a nuestras familias por todo el apoyo durante la formación académica esencial para lograr cumplir uno de nuestros sueños para lograr un futuro mejor y a nuestros amigos por sus ánimos continuos.

AGRADECIMIENTOS

Les damos las gracias a Dios sobre todas las cosas por dejarnos llegar hasta aquí, y a todos aquellos que durante el proceso siempre estuvieron a nuestro lado brindándonos apoyo, ánimos, fortaleza y haciendo de nosotros mejores personas capaces de cumplir con nuestros objetivos sin desviarnos del camino.

Muchas Gracias, por permanecer a nuestro lado.

TABLA DE CONTENIDO

Pág.

RESUMEN 9
ABSTRACT 10
INTRODUCCION 11
1. FASE DE DEFINICIÓN, PLANEACIÓN Y ORGANIZACIÓN 12
1.1. TITULO 12
1.2. TEMA 12
1.3. PLANTEAMIENTO 12
1.3.1. Descripción 12
1.3.2. Formulación 13
1.4. ALCANCES - DELIMITACIONES 13
1.4.1. Alcances 13
1.4.2. Delimitaciones 14
1.5. OBJETIVOS 14
1.5.1. General 14
1.5.2. Específicos 14
1.6. JUSTIFICACION 15
1.7. MARCO TEORICO 15
1.7.1. CRM 15
1.7.2. Teoría General de Sistemas 17
1.7.3. Gestión de Información 17
1.7.4. Colaboración 19
1.7.5. Herramientas 20
1.8. MARCO CONCEPTUAL 24
1.9. MARCO METODOLOGICO 25
1.10. FACTIBILIDAD 26
1.10.1. Económica 26
1.10.2. Técnica 27
1.10.3. Operativa 28
1.10.4. Legal 28

1.11. CRONOGRAMA 29
2. FASE DE ANALISIS 30
2.1. ADAPTABILIDAD A DISPOSITIVOS 30
2.2. SEGURIDAD 30
2.3. REQUERIMIENTOS 31
2.3.1. Tipos de Usuario 31
2.3.2. Pruduct Backlog 31
2.4. HISTORIAS DE USUARIO 32
2.5. TAREAS 37
2.6. SPRINTS 40
2.6.1. Iteración 1 40
2.6.2. Iteración 2 41
2.6.3. Iteración 3 43
2.6.4. Iteración 4 44
2.6.5. Iteración 5 45
2.6.6. Iteración 6 47
3. FASE DE DESARROLLO 48
4. CONCLUSIONES 93
5. RECOMENDACIONES 94
BIBLIOGRAFIA 95

LISTA DE TABLAS

Pág.

Tabla 1. Recursos....26
Tabla 2. Recursos de software....26
Tabla 3. Requerimientos de hardware....27
Tabla 4. Requerimientos de software....27
Tabla 5. Cronograma....29
Tabla 6 Requerimientos....31
Tabla 7 Tareas....37
Tabla 8 Pila Iteración 1....40
Tabla 9 Pila Iteración 2....41
Tabla 10 Pila Iteración 3....43
Tabla 11 Pila Iteración 4....44
Tabla 12 Pila Iteración 5....46

LISTA DE GRAFICAS

Pág.

Grafica 1 Ciclo de gestión de información 18
Grafica 2 Plantilla de entrada....48
Grafica 3 Componentes Formulario login 48
Grafica 4 Tabla User....49
Grafica 5 Formulario login....50
Grafica 6 Formulario recuperación....51
Grafica 7 Formularios excepciones....52
Grafica 8 Plantilla interna y menú responsivo....52
Grafica 9 Tabla permission, rol y sus relaciones....54
Grafica 10 Plantilla tabla....55
Grafica 11 Roles....56
Grafica 12 Formulario cambio permiso....57
Grafica 14 Permisos completos....58
Grafica 13 Permisos parciales....58
Grafica 15 Tabla log y sus relaciones....58
Grafica 16 Plantilla Formulario....60
Grafica 17 Perfil de usuario....61
Grafica 18 Formulario modificación datos personales....62
Grafica 19 Formulario excepciones....64
Grafica 20 Tabla payment y sus relaciones....64
Grafica 21 Plantilla ingreso....65
Grafica 22 Validación de pagos....66
Grafica 23 Tabla business, contact y sus relaciones....67
Grafica 24 Plantilla control clientes y prospectos....68
Grafica 25 Formularios control cliente y prospecto....70
Grafica 26 Plantilla de información....71
Grafica 27 Asociación de productos....73
Grafica 28 Tabla product y sus relaciones....74
Grafica 29 Plantilla control de productos....75
Grafica 30 Formulario agregar producto....76
Grafica 31 Plantilla administración....77
Grafica 32 Formularios supervisión de usuarios....78
Grafica 33 Plantilla información usuario....79
Grafica 34 Barra de Búsqueda....82
Grafica 35 Resultados búsqueda....83

Grafica 36 Formulario envió de correo.....84
Grafica 37 Correo satisfactorio 85
Grafica 38 Tabla record y sus relaciones.....85
Grafica 39 Seguimiento cliente 86
Grafica 40 Marketing.....87
Grafica 41 Correo satisfactorio 2 88
Grafica 42 Diagrama Circular clientes 90
Grafica 43 Tabla clientes asignados.....90
Grafica 44 Diagrama circular productos.....91
Grafica 45 Tabla productos vendidos 91

RESUMEN

Se realizó un levantamiento de información, en el área de ventas de tal forma elaborar un análisis identificando los sectores a mejorar y módulos a desarrollar. El aplicativo es utilizado la empresa “Mi-Martinez S.A.S” para el mejoramiento del manejo de vendedores, clientes y productos; y para el continuo envió de promociones.

El aplicativo permite el seguimiento de clientes, conocer sus fechas de pago y productos adquiridos, el registro de productos, asignación de productos y fechas de pago así como el precio variable dependiendo de las promociones de la empresa, el registro y seguimiento de administradores y vendedores así como su manejo y la asignación de clientes a los mismos, la asignación de permisos por tipo de usuario, un formulario de contacto con el supervisor encargado y uno para él envió de promociones a los clientes.

Los datos son ingresados por los usuarios, de forma tal poder proporcionar reportes de los clientes en el sistema su encargado y los productos adquiridos, el de los productos y sus ventas. Otorgando así un continuo seguimiento en los movimientos de cada usuario y clientes de la empresa.

PALABRAS CLAVE: clientes, manejo, mejorar, reportes, seguimiento, promociones.

ABSTRACT

Gathering information was held in the sales area so prepare an analysis identifying the sectors to improve and develop modules. The application is used the company "Mi-Martinez S.A.S" for improving the management of vendors, customers and products; and for continued promotions he sent.

The application allows tracking of customers, meet their payment dates and acquired products, product registration, product allocation and payment dates as well as the variable price depending on the developments of the company, registration and monitoring of managers and salespeople as well as their management and customer allocation to, the allocation of permits by type of user, a form of contact with the supervisor and one responsible for sending promotions to customers.

Data is entered by users, so as to provide reports of their customers in the system manager and the acquired products, products and sales. Thus providing continuous monitoring of movements of each user and enterprise customers.

KEYWORDS: customers, handling, improve, reports, monitoring, and promotions.

INTRODUCCION

Este trabajo de grado está basado en el estudio de ineficiencia de gestión de clientes y el control sobre los vendedores de la empresa Mi-Martinez S.A.S el cual es de propiedad y desarrollado por la misma. El estudio se basó en la toma de información de los productos adquiridos por cada cliente y control ejercido por los encargados de proporcionar información para incentivar a la compra de más productos de la empresa.

Después del análisis y verificación de los datos se llegó a la conclusión de que este proceso no cumplía con las expectativas de la empresa. Por lo que el proceso podía mejorarse con una plataforma que reuniera la información la compactara e entregara la misma a cada parte involucrada por lo que por pedido de la empresa se desarrollara en este trabajo.,

1. FASE DE DEFINICIÓN, PLANEACIÓN Y ORGANIZACIÓN

1.1. TITULO

SISTEMA DE INFORMACION WEB PARA LA GESTION DE CLIENTES (CRM) PARA LA EMPRESA MI-MARTINEZ

1.2. TEMA

El software a desarrollar se encargara del manejo de información de forma eficiente para la consulta e implementación del mismo, para este proceso se deberá tener información de cómo la empresa maneja y administra la información de sus clientes. Para mejorar el proceso del manejo con los clientes.

1.3. PLANTEAMIENTO

1.3.1. Descripción

El funcionamiento de la gestión de clientes en la empresa “Mi-Martinez S.A.S”, comienza cuando hay contacto con una persona que desea adquirir o cotizar algunos de los productos de la empresa “Mi-Martinez S.A.S” por medio telefónico o correo y los datos de ese contacto se agregan manualmente por un vendedor en el CRM (gestión de clientes) gratuito, después se les envía una cotización dándole un seguimiento de cómo va el estado de ese nuevo contacto, pero este sistema ofrece servicios muy limitados, como agregar el nombre, apellido, móvil, compañía, correo, etc. Pero hay más datos que la empresa desea guardar de los contactos como el cargo o poder vincular si hay más contactos en una misma compañía. Una de las limitaciones más recurrentes es la falta de interacción con los clientes, lo que evita llevarles un historial de seguimiento para poder ejecutar el trabajo correspondiente según cada tipo de cliente y el estado de su proceso en la empresa, además necesitan que se les vaya informando cuando es que los clientes tienen que pagar y así categorizar los clientes. Otra de las limitaciones es que no permite agregar vendedores para visualizar que clientes tienen los vendedores a cargo.

Todo esto conlleva que el proceso sea más largo y complicado ya que les toca agregar en una hoja de Excel los clientes que ya tienen un contrato con ellos y agregar manualmente en el Excel los pagos que son realizados por los clientes y agregar en otra hoja de Excel los contactos que apenas están cotizando y están en

proceso de contrato, pero no hay una forma de saber que vendedor esta encargados de los clientes.

1.3.2. Formulación

¿El desarrollo de un sistema de información web que permitirá gestionar los clientes de la empresa “Mi-Martinez S.A.S” de manera más eficiente?

1.4. ALCANCES - DELIMITACIONES

1.4.1. Alcances

El presente proyecto desarrolla un software, que incorpora una base de datos, gestión de información, facilidad de accesibilidad, enfocado a mejorar el rendimiento en cuanto a administración de usuarios en la empresa Mi-Martinez S.A.S para agilizar el proceso de gestión de clientes. Por lo que el aplicativo llevara los siguientes módulos:

- **Búsqueda:** Se brinda la información requerida por el administrador o vendedor.
- **Gestión de Clientes:** Gestiona la información de los clientes que a su vez contendrá un histórico de cambios por cada cliente y proporcionara un sistema de búsqueda.
- **Marketing:** Proporciona la facilidad de enviar mensajes a determinados usuarios.
- **Registro:** Gestiona el ingreso de administradores y vendedores, controla el acceso de los mismos y da los permisos pertinentes a cada uno.
- **Seguridad:** proporciona y restringe privilegios a los diferentes tipos de usuarios.
- **Ventas:** Gestiona la información de los producto que proporciona la empresa y las historias de ventas.

1.4.2. Delimitaciones

Geográfica: El software pretende organizar y optimizar el proceso de gestión de clientes en la empresa “Mi-Martinez S.A.S”. El proyecto es desarrollado en la sede de la empresa.

Temporal: La implementación del proyecto está proyectada, dentro de seis (6) meses.

Técnica: Las especificaciones y entornos en los cuales se desarrolla el software son las siguientes:

- Sistema operativo Linux y Windows.
- Lenguaje de entorno Web HTML, JavaScript, PHP, CSS.
- Servidor Web.
- Motor de Base de datos MySQL.
- Entorno de desarrollo Sublime Text.
- Framework Laravel.

1.5. OBJETIVOS

1.5.1. General

Desarrollar un sistema de información web para la gestión de clientes (CRM) para la empresa “Mi-Martinez S.A.S”.

1.5.2. Específicos

- Utilizar métodos web de adaptabilidad para el buen funcionamiento del sistema en diferentes dispositivos.
- Desarrollar los módulos de los sectores de registro, búsqueda, seguridad, ventas, gestión de clientes y marketing para agilizar y optimizar los procesos CRM de la empresa.
- Diseñar la Base de Datos para organizar la información del sistema.
- Implementar algoritmos de seguridad proporcionados por LARAVEL para obtener un aplicativo seguro.

1.6. JUSTIFICACION

Teniendo en cuenta la gran cantidad de clientes que se manejan dentro de la empresa “Mi-Martinez”, se muestra la necesidad de un sistema de manejo y control de los mismos que permita la búsqueda, ingreso e historias de usuario de una forma eficiente; capaz de reducir los tiempos en los procesos que requieren conocer e ingresar clientes. Con este propósito se plantea la creación de una aplicación para mejorar la calidad del trabajo, agilizar los procesos de gestión de clientes y dar efectividad tanto a “Mi-Martinez S.A.S” como a los usuarios que utilizaran esta aplicación, siendo la mejor forma de dar solución al procesamiento y almacenamiento de información mientras se adopta un revestimiento de las herramientas digitales que dan proximidad a excelentes procesos de trabajo.

El uso del software permitirá que los administradores y vendedores sean capaces de llevar una base de datos controlada con cada movimiento que el cliente haga, llevando así un historial, calendario, y permitiendo saber en qué áreas está interesado, para poder abordarlo y enviarle mensajes relacionados con sus intereses de tal forma incentivarlo a comprar más de los producto que ofrece la empresa. También incluirá un sector de ventas que mostrara que productos son los más requeridos y que tipo de clientes los solicitan para poder abordar las falencias y entender que nuevos productos producir y cuales retirar de su inventario con el fin de mejorar cada vez más.

Adicionalmente, con un sistema de registro y algoritmos de seguridad que se implementara se garantizara la total seguridad del software manteniendo la privacidad e integridad de la empresa sin ningún tipo de perdida información que le dará a la misma una total garantía de control y privacidad sobre sus movimientos en el control de clientes.

1.7. MARCO TEORICO

1.7.1. CRM

Constituye un proceso de incremento de valor apoyado por las tecnologías de la información, que identifica, desarrolla, integra y orienta las distintas competencias de la empresa hacia la voz de los clientes, con objeto de entregar un mayor valor al cliente en el largo plazo, para identificar correctamente los segmentos de mercado tanto existentes como potenciales.

ENFOQUE TECNOLÓGICO

Fruto del papel fundamental que desempeñan las TI como factor posibilitador de dicha estrategia, encontramos diversos términos y delimitaciones que hacen referencia al ámbito tecnológico de la misma. Nos ha parecido interesante incluir una breve lista con dichos términos, con objeto de profundizar en el análisis del concepto de CRM y en sus diversos matices. Desde un enfoque tecnológico, podemos considerar los siguientes términos:

- **Sistemas o soluciones CRM:** sistemas informáticos que apoyan la gestión de relaciones con clientes (Xu y Walton, 2005). - E-CRM: hace referencia a la utilización de Internet o la web como canal preferente para el establecimiento y desarrollo de relaciones con los clientes (Xu y Walton, 2005). Dichos autores insisten en que los sistemas e-CRM permiten, gracias al uso de Internet, la disponibilidad de información sobre clientes a lo largo de todos los puntos de contacto dentro de la compañía y con los socios externos a través de intranets. Por otra lado, (Racherla y Hu, 2006) definen e-CRM como CRM electrónico o basado en la web (Web based electronic customer relationship management).

- **Mobile CRM:** estrategia CRM que incluye comunicación unidireccional o interactiva entre la empresa y el cliente utilizando un dispositivo móvil, lo que ofrece una mayor autonomía temporal y espacial a la hora de establecer la comunicación (Sinisalo et al., 2007; Liljander, Polsa y Forsberg, 2007). Los dispositivos móviles incluyen distintas herramientas de comunicación móvil inalámbricas: teléfono móvil, teléfono inteligente, PDA y tagers (buscapersonas).

ENFOQUE DEL CRM COMO FILOSOFÍA DE NEGOCIO

Dejando a un lado el aspecto tecnológico, otro término que nos ha parecido interesante incluir en esta lista ha sido la consideración del CRM como filosofía de negocio, ya que es una denominación que encontraremos en multitud de trabajos y estudios.

Filosofía CRM: la consideración de CRM como filosofía de negocio está intrínsecamente unida al concepto de marketing, ya que para conseguir la lealtad del cliente, la empresa debe estar realmente orientada al mismo, dando respuesta a sus necesidades. Por tanto, dicha perspectiva enfatiza la importancia de la creación de valor: para que el CRM sea efectivo y las relaciones con clientes sean duraderas, la empresa debe descubrir continuamente qué es lo que sus clientes valoran e incorporarlo a sus productos y servicios (Zablah, Bellenger y Johnston, 2004).

No podemos finalizar el presente epígrafe dedicado al análisis del concepto de CRM sin comentar que, como hemos observado en el análisis realizado, no existe aún

una conceptualización común o delimitación conceptual generalmente aceptada sobre CRM. En este sentido, si nos centramos en la estructura básica o marco conceptual de CRM, expertos e investigadores han establecido diversas perspectivas, sin llegar a un consenso al respecto (Cheng, Chang y Liu, 2005).

1.7.2. Teoría General de Sistemas

Es un método: que nos permite unir y organizar los conocimientos con la intención de una mayor eficacia de acción.

Engloba la totalidad de los elementos del sistema estudiado así como las interacciones que existen entre los elementos y la interdependencia entre ambos.

El objetivo de la teoría es la descripción y exploración de la relación entre los sistemas dentro de esta jerarquía. Hay que distinguir "sistema" de "agregado". Ambos son conjuntos, es decir, entidades que se constituyen por la concurrencia de más de un elemento; la diferencia entre ambos consiste en que el sistema muestra una organización de la que carecen los agregados. Así pues, un sistema es un conjunto de partes interrelacionadas[1].

Los sistemas pueden ser:

SISTEMA ABIERTO: Relación permanente con su medio ambiente. Intercambia energía, materia, información. Interacción constante entre el sistema y el medio ambiente.
SISTEMA CERRADO: Hay muy poco intercambio de energía, de materia, de información, etc. Con el medio ambiente. Utiliza su reserva de energía potencial interna.

1.7.3. Gestión de Información[2]

La gestión de información en actividades relacionadas a la acción integral contra minas antipersonal, debe ser vista como un recurso meritorio y estratégico para la toma y consolidación de las decisiones, teniendo en cuenta que la información debe estar enmarcada en una visión estructurada y encaminada al logro efectivo, eficiente y eficaz de las metas, objetivos y resultados de las instancias de la acción contra

[1]Aniorte, teoría general de sistema[en línea], [citado en 22 agosto de 2014], <http://www.aniorte-nic.net/apunt_terap_famil_2.htm>
[2] Gestión de la información[en línea], [citado en 22 agosto de 2014], <http://www.accioncontraminas.gov.co/accion/Paginas/GestionInformacion.aspx>

minas u otra organización. En este sentido, podemos entender que la gestión de información debe ser concebida como un proceso que involucra actividades de: *tareas de recolección, almacenamiento, tratamiento depuración y reporte de datos*, provenientes por diferentes fuentes que gestionan el acceso sobre la misma, se establece entonces, como una DISCIPLINA TRANSVERSAL, que entrelaza todas la capas o tejidos de una organización.

Como todo proceso, es necesario realizar un estudio previo de las necesidades de información, lo cual permitirá determinar con exactitud qué información se necesita para apoyar las decisiones, lograr la estandarización de los conceptos y el escalonamiento del ciclo de gestión de información, este último será la herramienta para poder estructurar el éxito de la misma, un ciclo permite obtener información sobre los resultados obtenidos, e introducir modificaciones a lo largo de su aplicación. Ante todo, es contar con un plan estratégico en el que se indique el modo en que se utilizarán los recursos para atender las futuras necesidades en materia de información.

Al organizar y coordinar las actividades de gestión de la información en su ciclo, es preciso tomar en consideración las políticas y el horizonte del que hacer, por cuanto es importante definir las fuentes, los métodos para la recopilación de datos, la sistematización y uso de la herramienta tecnológica y ello surtir un proceso de aprobación.

Por último, los datos que no se utilizan ni difunden no son útiles. La pertinencia, la actualidad y la precisión son los tres principios que deben orientar la utilización y recopilación de la información.

CICLO DE GESTIÓN DE INFORMACIÓN



Grafica 1 Ciclo de gestión de información

Planificación de la gestión de la información
Recopilación de datos: fuentes
Recopilación de datos: métodos
Recopilación de datos: pruebas
Supervisión de la recopilación de datos
Conservación de los datos
Utilización de los datos
Difusión de los datos

ESTRATEGIA DE DESCENTRALIZACIÓN DE INFORMACIÓN

El objetivo central de la gestión de información en torno a cualquier problemática es generar conocimiento profundo y sistemático tanto de la situación como de todas aquellas acciones para contribuir a su solución, de tal manera que permitan generar los análisis correspondientes para apoyar la toma de decisiones en cuanto a futuras acciones así como apoyo a la generación o modificación de políticas. En este sentido, la información se convierte en la memoria de la problemática.

La gestión de información se refiere a la recolección, verificación, registro, validación, procesamiento y manejo de información necesaria a partir de una o más fuentes y su distribución a uno o más usuarios que tienen interés o derecho a esa información. En este sentido la gestión de información apoya el cumplimiento de las obligaciones del PAICMA en lo estipulado en la ley 759 de 2002 y el Decreto 2150 de 2007, en cuanto a:

- Mantener la base del Sistema de Información de Acción Contra Minas, encargándose de recopilar, sistematizar, centralizar y actualizar toda la información sobre el tema (Función numeral 6 artículo 2, decreto 2150 de 2007).

- Servir de fuente para la toma de decisiones de acuerdo con la información recolectada sobre los programas de prevención, señalización, elaboración de mapas, remoción de minas y atención a víctimas (Función numeral 7 artículo 2, decreto 2150 de 2007).

- Centralizar la información de todas las organizaciones que desarrollen actividades relativas a minas y consolidar todos los datos que estas recolecten mediante las actividades que desarrollen (Función numeral 8 artículo 2, decreto 2150 de 2007).

1.7.4. Colaboración

Hasta el latín hay que marcharse para poder encontrar el origen etimológico del término colaboración que ahora vamos a analizar en profundidad. En concreto, aquel se encuentra conformado por los siguientes componentes léxicos latinos: el prefijo “con-“, que es equivalente a “junto”; el verbo “laborare”, que puede traducirse como “trabajar”, y el sufijo “-ción”, que es sinónimo de “acción y efecto”.

Colaboración es la **acción y efecto de colaborar**. Este verbo refiere a trabajar en **conjunto** con otra u otras **personas** para realizar una obra. Por ejemplo: *“Gracias a la colaboración de los asistentes, la cena benéfica ha recaudado más de veinte mil pesos”*, *“¿Puedo pedir tu colaboración? Necesito usar está*

máquina y no sé cómo hacerlo", *"Nunca hubiese podido empujar el camión sin la colaboración de los policías"*.

La colaboración, por lo tanto, es una *ayuda* que se presta para que alguien pueda lograr algo que, de otra manera, no hubiera podido hacer o le hubiera costado más. En este sentido, se puede colaborar con una anciana que necesita subir una escalera o con un no vidente que tiene que cruzar la calle.

Dentro del ámbito de la música, es habitual que se use también el término colaboración que ahora nos ocupa. En concreto, aquel se suele emplear para referirse a que un cantante lanza al mercado una composición que es interpretada a dúo con otro artista. A lo largo de la Historia muchos son los profesionales de la canción que han recurrido a colaboraciones para darle mayor relevancia a su álbum discográfico o simplemente para otorgarle mayor reconocimiento a una de sus composiciones. Ejemplos de colaboraciones musicales que han conseguido gran éxito han sido la de Beyoncé con Lady Gaga para la canción "Telephone"; la de Marta Sánchez con Andrea Bocelli para "Vivo por ella"; la de Alejandro Sanz con Shakira para "La tortura" o la de Michael Jackson y Van Halen para "Beat it".[3]

1.7.5. Herramientas

- **HTML5**[4]

Es la última evolución de la norma que define HTML. El término representa dos conceptos diferentes. Se trata de una nueva versión del *lenguaje* HTML, con nuevos elementos, atributos y comportamientos, y un conjunto más amplio de tecnologías que permite a los sitios Web y las aplicaciones más diversas y de gran alcance. Este conjunto se le llama *HTML5 y amigos* y, a menudo reducido a sólo *HTML5*.
Diseñado para ser utilizable por todos los desarrolladores de Open Web, esta referencia página enlaza numerosos recursos sobre las tecnologías de HTML5, que se clasifican en varios grupos según su función.

- *Semántica*: lo que le permite describir con mayor precisión cuál es su contenido.
- *Conectividad*: lo que le permite comunicarse con el servidor de formas nuevas e innovadoras.
- *Desconectado y almacenamiento*: permite a páginas web almacenar datos, localmente, en el lado del cliente y operar fuera de línea de manera más eficiente.

[3] Colaboración [en línea], [citado en 22 de septiembre de 2013]< http://definicion.de/colaboracion/ >
[4] Mozilla, HTML5[en línea], [citado 22 agosto de 2014], <https://developer.mozilla.org/es/docs/HTML/HTML5>

- *Multimedia*: permite hacer vídeo y audio de ciudadanos de primera clase en la Web abierta.
- *Gráficos y efectos 2D/3D*: permite una gama mucho más amplia de opciones de presentación.
- *Rendimiento e Integración*: proporcionar una mayor optimización de la velocidad y un mejor uso del hardware del equipo.
- *Dispositivo de Acceso*: admite el uso de varios dispositivos de entrada y salida.
- *Styling*: deja a los autores escribir temas más sofisticados.

➢ PHP 5.5[5]

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

¿Cómo trabaja PHP? El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección tipo http://www.aprenderaprogramar.com/index.php en un navegador web como Internet Explorer, Firefox o Chrome, ¿qué ocurre? Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

El esquema es:

Petición de página web al servidor --> El servidor recibe la petición, reúne la información necesaria consultando a bases de datos o a otras páginas webs, otros servidores, etc --> El servidor responde enviando una página web “normal” (estática) pero cuya creación ha sido dinámica (realizando procesos de modo que la página web devuelta no siempre es igual).

[5]Enrique Gonzales, PHP[en línea], [citado 22 agosto de 2014],<http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193>

En resumen:

Páginas estáticas: Petición --> Respuesta
Páginas dinámicas: Petición --> Procesado y preparación --> Respuesta

En un sitio dinámico, la información generalmente está contenida en una base de datos. Cada vez que mostramos la página, como por ejemplo una página de noticas, buscamos en la base de datos las últimas noticias que tenemos ingresadas para mostrar en el navegador del visitante. Ahora bien, ¿cómo se consigue que la página vaya mostrando noticias nuevas? Simplemente cargando las mismas en la base de datos, por ejemplo, a través de un formulario que rellena una persona y que una vez relleno cuando pulsa "Enviar" implica que lo que haya escrito se guarde en nuestra base de datos. De esta manera, cuando solicitamos la página web y el servidor consulte la base de datos, encontrará esta nueva información y mostrará nuestra página de forma distinta a como se veía anteriormente.

- **JavaScript**

JavaScript es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web.

Para entender lo que es JavaScript consideremos lo siguiente. Un usuario escribe una dirección web en su navegador, por ejemplo http://www.aprenderaprogramar.com. El servidor recibe la petición y como respuesta a esa petición envía al ordenador del usuario código HTML junto a código JavaScript. El código HTML se encarga de que en la pantalla se muestre algo, por ejemplo una imagen, un menú, etc. El código JavaScript se puede encargar de crear efectos dinámicos en respuesta a acciones del usuario, por ejemplo que se despliegue un menú tipo acordeón cuando el usuario pasa el ratón por encima de un elemento del menú.

La ventaja de JavaScript es que al estar alojado en el ordenador del usuario los efectos son muy rápidos y dinámicos. Al ser un lenguaje de programación permite toda la potencia de la programación como uso de variables, condicionales, bucles, etc. También podemos citar algún inconveniente: por ejemplo si el usuario tiene desactivado JavaScript en su navegador, no se mostrarán los efectos. No obstante, hoy día la mayoría de los usuarios navegan por la web con JavaScript activado.

FORMAS DE USAR JAVASCRIPT EN DOCUMENTOS HTML

La primera forma de usar JavaScript dentro de una página web es embebiendo directamente el código JavaScript dentro del código HTML. Vamos a poner un ejemplo de código JavaScript pero no vamos a entrar en detalle a comentarlo ni explicarlo ya que en este curso no vamos a explicar JavaScript. Simplemente queremos saber interpretar por qué dentro de un documento HTML pueden aparecer fragmentos de código escrito en otro lenguaje.

- **Servidor Web**

Es la maquina o computador donde se almacena su página web. Toda la información publicada en cada sitio web se almacena en un espacio destinado para este fin. De lo contrario no habría forma de divulgar el contenido.

¿Cómo funciona el servidor web?

Para que el servidor web funcione correctamente se debe utilizar un método para intercambiar la información o transferir los sitios web al ordenador, esto se hace a través del protocolo http **HyperText Transfer Protocol** (Protocolo de transferencia de hipertexto) que es el que se refleja antes de escribir cualquier dirección de internet. Por ejemplo: **http://www.empresamia.com**

¿Cuál es el servidor web más utilizado?

Gracias a que en la actualidad existe la banda ancha, hoy es posible que las empresas utilicen servidores web más rentables. Es el caso de Apache, el servidor web más utilizado debido a la implementación de protocolos actualizados y a la rapidez con que muestra la información. Apache además cuenta con los constantes aportes de un grupo de voluntarios que trabaja en el mejoramiento del mismo y tiene el soporte de la Fundación Apache, la cual vela por ofrecer un software de alta calidad, así como el respaldo de distintas empresas que generan versiones pagas del popular servidor. Además de Apache, también existen otros servidores como ISS, Cherokee, Monkey http, Lighttpd y Thttpd.

Primero Usted deberá primero decidir si tiene el servicio de hosting en un servidor web propio o ajeno. Y segundo, la rapidez y capacidad de de su servidor web serán factores determinante en el éxito de su página web y el número de visitas que reciba. [6]

[6] Empresamia, servidor web [en línea], [citado en 22 de octubre de 2013] <http://empresamia.com/crear-empresa/crear/item/644-que-es-un-servidor-web>

1.8. MARCO CONCEPTUAL

CRM

Consiste en una estrategia de la organización en la cual centra sus esfuerzos en el conocimiento de sus clientes, detectando sus necesidades, aumentando su grado de satisfacción, incrementando su fidelidad a la empresa e incrementando la rentabilidad o beneficios del cliente a la empresa, mediante el análisis de las informaciones extraídas por los clientes desde los diferentes canales o medios de comunicación.[7]

DATOS

Son la materia en bruto de la que se parte. No tienen contexto a no ser que se los interprete.[8]

GESTION

Se trata de realización de diligencias enfocadas a la obtención de algún beneficio, tomando a las personas que trabajan en la compañía como recursos activos para el logro de los objetivos. [9]

INFORMACIÓN

Son el conjunto de datos organizados y presentados de una determinada manera, y revela el significado de estos, o por lo menos, una interpretación.[10]

MARKETING

Es un conjunto de actividades destinadas a satisfacer las necesidades y deseos de los mercados meta a cambio de una utilidad o beneficio para las empresas u organizaciones que la ponen en práctica; razón por la cual, nadie duda de que el marketing es indispensable para lograr el éxito en los mercados actuales.[11]

---

[7] CRM - Customer Relationship Management. [en línea],[citado en 22 octubre de 2013] <http://www.webandmacros.com/crm.htm>

[8]Construcción y cristalización del conocimiento [en línea],[citado en 22 octubre de 2013] <http://www.elperiodicodearagon.com/noticias/idear/construccion-y-cristalizacion-del-conocimiento_215219.html>

[9] Definición de gestión [en línea],[citado en 22 octubre de 2013] <http://www.gestionyadministracion.com/empresas/definicion-de-gestion.html>

[10]Construcción y cristalización del conocimiento [en línea],[citado en 22 octubre de 2013] <http://www.elperiodicodearagon.com/noticias/idear/construccion-y-cristalizacion-del-conocimiento_215219.html>

[11] Definición Marketing [en línea],[citado en 22 octubre de 2013] <http://www.marketing-free.com/marketing/definicion-marketing.html>

SEGURIDAD

Es una disciplina que se encarga de **proteger la integridad y la privacidad** de la **información** almacenada.[12]

1.9. MARCO METODOLOGICO

Para el desarrollo de este trabajo, se plantean la metodología SCRUM con las siguientes etapas:

- Etapa 1 (Exploración): Análisis de bibliografía existente referida a procesos productivos en entornos Ágiles, a modelos/normas y estándares de calidad y mejora de procesos, Control estadístico de procesos.

- Etapa 2 (Análisis): Análisis de implementaciones actuales de prácticas de alta madurez en entornos de desarrollo basados en Scrum.

- Etapa 3 (Experimentación): Desarrollo del proceso definido para la gestión cuantitativa de proyectos en entornos de desarrollo utilizando Scrum. Diseño de la herramienta que permita la automatización de la generación de cartas de control a partir de los datos obtenidos de la aplicación de gestión de proyectos basados en Scrum.

- Etapa 4 (Implementación): Validación del proceso definido propuesto, y definición de elementos para la posterior generación de e-learning.

- Etapa 5 Conclusiones finales del trabajo.

El trabajo se desarrollara de la siguiente forma:

Sprint Planning (cada 60 días):

1. Timebox de cómo máximo 4 horas con el cliente.
2. Timebox de cómo máximo 4 horas entre el equipo.

Sprint (durante 30 días o menos dependiendo de la eficiencia del equipo):

[12] Seguridad Informática [en línea],[citado en 22 octubre de 2013] <http://definicion.de/seguridad-informatica/>

- Reunión de sincronización (todos los días de trabajo hábiles máximo 15 minutos)
- Realización de cada tarea por horas
- Unión de todos los sistemas desarrollados

1.10. FACTIBILIDAD

1.10.1. Económica

El proyecto implicara los siguientes recursos presentados en la Tabla 1:

Tabla 1. Recursos

RECURSO	DESCRIPCION	CANTIDAD	HORAS MES	TIEMPO TOTAL	VALOR HORA	COSTO MES	VALOR TOTAL
Humano	Analista Diseñador Ingeniero de desarrollo	2	80	480	$ 12.000	$ 960.000	$ 11.520.000
	Asesor	1	4	24	$ 25.000	$ 100.000	$ 600.000
Hardware	Computador	2	120	720	$ 500	$ 60.000	$ 720.000
	Servidor	1	720	4320	$ 1.000	$ 720.000	$ 4.320.000
Otros	Papelería					$ 30.000	$ 180.000
	Internet					$ 70.000	$ 420.000
	Transporte					$ 70.000	$ 420.000
Total							$ 18.180.000

Autores. Edward Díaz, Manuel Pinzón

También, se implicaran los siguientes recursos de software expuestos en la Tabla 2:

Tabla 2. Recursos de software

ITEM	SOFTWARE	TIPO LICENCIA
Sistema Operativo	Windows	Propietario
Sistema Operativo	Linux	Libre
Base de Datos	MySQL	Libre
Servidor Web	Apache	Libre

Framework de desarrollo	Laravel	Libre
Servidor de Aplicaciones	Glassfish	Libre
Ambiente de Desarrollo	Sublime Text	Libre

Autores. Edward Díaz, Manuel Pinzón

El software privado o propietario será suministrado por la empresa “Mi-martinez S.A.S”, por ende el proyecto es factible.

1.10.2. Técnica

El proyecto requerirá de los componentes presentados en la Tabla 3 para su elaboración e implementación:

Tabla 3. Requerimientos de hardware

ITEM	**FUNCION**
Computador	Desarrollo del Aplicativo
Servidor	Almacenar y dar disposición del proyecto

Autores. Edward Díaz, Manuel Pinzón

Los computadores serán proporcionados por los estudiantes y la empresa, mientras que el servidor será proporcionado por la empresa “Mi-martinez S.A.S”, por ende el proyecto es factible.

El proyecto implica la consecución del software presentados en la Tabla 4, para su desarrollo y funcionamiento:

Tabla 4. Requerimientos de software

ITEM	SOFTWARE	FUNCION
Sistema Operativo	Windows, Linux	Almacena el proyecto
Base de Datos	MySQL	Almacena la información
Servidor Web	Apache	Comparte el proyecto a través de la red
Servidor de Aplicaciones	Glassfish	Gestionar las funciones de lógica de negocio y de acceso a los datos de la aplicación.
Ambiente de Desarrollo	Sublime Text	Desarrollo del aplicativo

Autores. Edward Díaz, Manuel Pinzón

1.10.3. Operativa

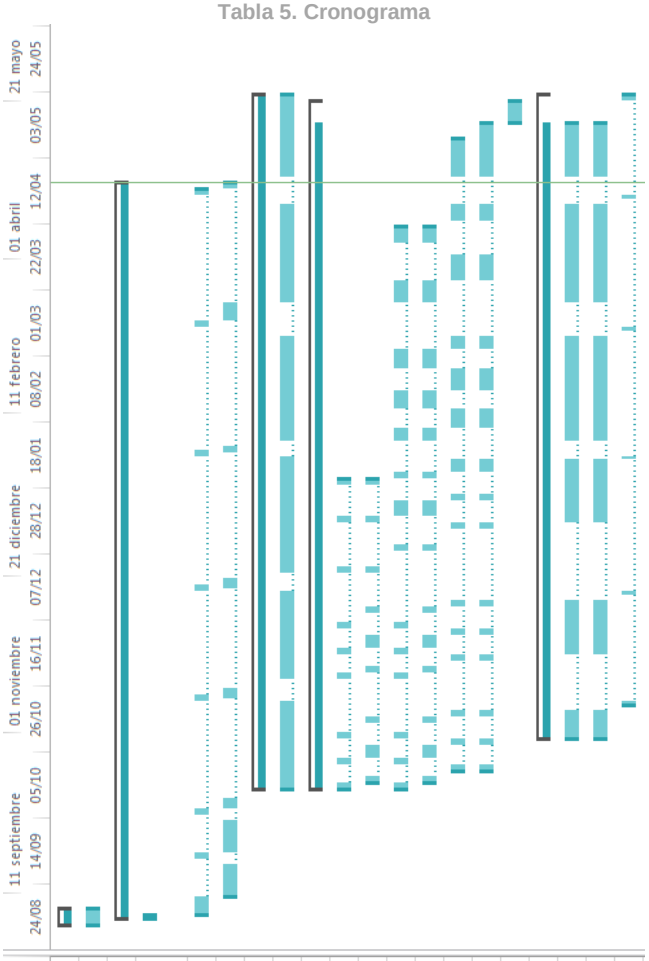
Una vez terminado el proyecto, el software va a ser utilizado por la empresa “Mi-martinez S.A.S”, posteriormente la empresa dispondrá de monitores que administraran el sistema.

1.10.4. Legal

El proyecto es factible legalmente porque cuenta con las licencias de sistema operativo y de ambiente de desarrollo de la empresa “Mi-martinez S.A.S”. Además, gracias a la “Ley de Propiedad Intelectual” o **Ley 11723** de Colombia el resto de software libre necesario puede ser implementado en el proyecto por lo tanto es factible legalmente.

1.11. CRONOGRAMA

Tabla 5. Cronograma



	Nombre de tarea	Duración	Comienzo	Fin
1	**Exploracion**	**5 días**	**lun 01/09/14**	**vie 05/09/14**
2	Planeacion	5 días	lun 01/09/14	vie 05/09/14
3	**Analisis**	**153 días**	**mié 03/09/14**	**vie 24/04/15**
4	Estructurar entorno de desarrollo	1 día	mié 03/09/14	mié 03/09/14
5	Levantamiento Informacion	18 días	jue 04/09/14	mié 22/04/15
6	Estructurar Sistemas	33 días	mié 10/09/14	vie 24/04/15
7	**Experimentacion**	**143 días**	**mar 14/10/14**	**vie 22/05/15**
8	Sprints	120 días	mar 14/10/14	vie 22/05/15
9	**Desarrollo Sistemas**	**141 días**	**mar 14/10/14**	**mié 20/05/15**
10	Sistema 1	18 días	mar 14/10/14	mar 20/01/15
11	Sistema 2	18 días	jue 16/10/14	mar 20/01/15
12	Sistema 3	39 días	mar 14/10/14	vie 10/04/15
13	Sistema 4	39 días	jue 16/10/14	vie 10/04/15
14	Sistema 5	51 días	lun 20/10/14	vie 08/05/15
15	Sistema 6	54 días	lun 20/10/14	mié 13/05/15
16	Configuracion Servidor	4 días	jue 14/05/15	mié 20/05/15
17	**Implementacion**	**131 días**	**jue 30/10/14**	**vie 22/05/15**
18	Pruebas	87 días	jue 30/10/14	mié 13/05/15
19	Cambios Necesarios	87 días	jue 30/10/14	jue 14/05/15
20	Conclusiones	7 días	lun 10/11/14	vie 22/05/15

2. FASE DE ANALISIS

2.1. ADAPTABILIDAD A DISPOSITIVOS

El diseño web adaptable o adaptativo, conocido por las siglas RWD del inglés *Responsive Web Design*, es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visualizarla.

- El diseño es elaborado a partir de la funcionalidad en dispositivos móviles y ampliado a medida que se pensaba como se comportaría en dispositivos de mayor resolución y tamaño.

- La adaptabilidad es implementada gracias al método @media proporcionado por CSS el cual nos permite determinar el ancho en pixeles (px) del dispositivo y navegador en uso, al mismo tiempo permite el cambio de parámetros y atributos de las etiquetas HTML cuando el dispositivo se encuentre en determinados pixeles ya establecidos.

- El código CSS es elaborado a partir del principio de “No más pixeles” lo que significa desarrollar el código sin estar consciente del tamaño fijo de los elementos visuales, por lo que se otorgan atributos de tamaño como porcentajes (%), rem, em, que son medidas proporcionales a elementos padres. En este proyecto aún se utilizan pixeles para determinar márgenes, fuentes, etc. Menores a 21px.

  El código implementado es expuesto en el desarrollo del proyecto a medida que el mismo avanza.

2.2. SEGURIDAD

- Se utiliza las librerías y métodos proporcionados por Laravel para la encriptación, consulta, actualizaciones y guardado de contraseñas cifradas.

- Las autentificaciones y sesiones también son proporcionados por Laravel.

- Todo formulario es creado a través del código {{Form::función}} esto es proporcionado por Laravel y su sistema de plantilla denominado Blade que nos permite la combinación de código PHP y HTML en un mismo archivo de esta forma toda la información del formulario es entregada al servidor a través del método POST.

- Las rutas son elaboradas con el código Route::Método (GET, POST, PUT, DELETE) lo cual permite la restricción de acceso a los usuarios por métodos diferentes de los antes mencionados.

- Laravel proporciona un clase llamada filter, estos filtros son funciones por las cuales tiene que pasar una ruta antes de ser entregada al usuario final, en ellas se colocan funciones de verificación de permisos y demás comprobaciones, en las cuales si no se cumple con lo establecido son re direccionados o se muestra una página diferente a la solicitada.

La implementación de estos métodos y el código utilizado es expuesta en el desarrollo del proyecto.

2.3. REQUERIMIENTOS

2.3.1. Tipos de Usuario

Súper Administrador: Este tipo de usuario tiene acceso total de la plataforma, teniendo permisos de crear, modificar y eliminar sin ninguna restricción.

Administrador: Este tipo de usuario tiene acceso restringido de la plataforma, teniendo permisos de crear y modificar.

Vendedor: Este tipo de usuario tiene un acceso restringido a la plataforma y es un usuario estándar con unos permisos específicos.

2.3.2. Pruduct Backlog

Tabla 6 Requerimientos

PRIORIDAD	DESCRIPCION	ESTIMACION
1	Se requiere que la plataforma sea capaz de funcionar en cualquier dispositivo de forma óptima y visualizarse bien, además de ser simple y fácil acceso para cualquier usuario con o sin experiencia manejando plataformas.	40
2	Sistema de acceso el cual permita el ingreso y verificación del usuario a la plataforma, y él envió de la solicitud de recuperación de contraseña en caso de olvido.	30
3	Cada usuario debe tener un perfil, registro de actividades (logs), y permisos de acceso a determinados sectores de la plataforma; además de poder modificar su información personal.	50
4	Cada usuario debe poder ver al ingreso de la plataforma los clientes con pagos pendientes.	15

5	Se requiere de los sectores de control de clientes y prospectos, control de productos, supervisión de usuarios y administración de permisos. A estos sectores se podrá ingresar dependiendo de los permisos asignados a cada tipo de usuario.	30
6	Se requiere de un sistema de búsqueda general o por filtros para encontrar a los clientes, prospectos, empresa.	20
7	El usuario debe poder enviar mensajes masivos a sus supervisores en la plataforma.	15
8	Se requiere de un historial de los clientes para su supervisión.	20
9	Se requiere un sector de estructuración de correos, que permita programar correos masivos, y enviárselos a los clientes.	40
10	Se requiere de un sector de reportes el cual contendrá la cantidad de prospectos, clientes, recurrentes y la cantidad de clientes encargados a un vendedor	40

2.4. HISTORIAS DE USUARIO

1	**INGRESAR AL SISTEMA**	
Como usuario, quiero ingresar al sistema y en caso de olvido de contraseña poder recuperarla, para poder acceder a la funcionalidad del mismo.		
Estimación: 8		Riesgo: Medio
Prioridad: 500		Dependiente de:
Pruebas de Aceptación: • Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.		

2	**NAVEGACION EN EL SISTEMA**	
Como usuario, quiero poder navegar a través del sistema rápida y eficientemente, para realizar las tareas asignadas.		
Estimación: 4		Riesgo: Bajo
Prioridad: 300		Dependiente de:

Pruebas de Aceptación:

- Se re direcciona correctamente a todas las áreas en el sistema.

3	**DAR PERMISOS DE ACCESO**

Como súper-administrador, quiero determinar a qué zonas pueden o no tener acceso los administradores y vendedores, para una mayor seguridad.

Estimación:	6	Riesgo:	Alto
Prioridad:	600	Dependiente de:	

Pruebas de Aceptación:

- Intentar acceder a un sector no permitido.

4	**REGISTRAR ACTIVIDADES (LOGS)**

Como usuario, quiero llevar un registro de todos los movimientos propios y del usuario a cargo, para un mayor control.

Estimación:	4	Riesgo:	Bajo
Prioridad:	300	Dependiente de:	

Pruebas de Aceptación:

- Obtener los datos de una actividad realizada en la zona de movimientos.

5	**PERFIL**

Como usuario, quiero un sector donde pueda consultar, modificar mi información personal, para mayor seguridad.

Estimación:	6	Riesgo:	Bajo
Prioridad:	300	Dependiente de:	

Pruebas de Aceptación:

- Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.

6	**PAGOS PENDIENTES**

Como usuario, quiero observar como primera instancia los pagos pendientes de los clientes a mi cargo o de aquellos usuarios a mi cargo y poder actualizar, validar el pago, para un mayor control.

Estimación:	6	Riesgo:	Medio
Prioridad:	600	Dependiente de:	7, 8

Pruebas de Aceptación:

- Los clientes están a mi cargo.
- La validación es correcta.

7	**CONTROLAR CLIENTES Y PROSPECTOS**

Como usuario, quiero disponer de un sector con la lista de clientes y prospecto, el poder crear, modificar clientes y la asociación del mismo con los productos ofrecidos por la empresa y el usuario encargado, además de poder buscar a un cliente especifico.

Estimación:	6	Riesgo:	Alto
Prioridad:	500	Dependiente de:	8

Pruebas de Aceptación:

- Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.
- Asociación de cliente con el producto satisfactoria.

8	**CONTROLAR PRODUCTOS**

Como usuario, quiero poder agregar, consultar, eliminar, modificar productos, y observar la lista completa y por categorías, para un mayor control en el seguimiento de los mismos.

Estimación:	4	Riesgo:	Media
Prioridad:	400	Dependiente de:	

Pruebas de Aceptación:

- Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.

9	**SUPERVISAR USUARIOS**

Como súper-administrador y administrador, quiero disponer de un sector con la lista de usuarios, el poder crear, modificarlos y la asociación del mismo con los productos ofrecidos por la empresa, además de poder buscar a un usuario especifico.

Estimación:	6	Riesgo:	Medio
Prioridad:	300	Dependiente de:	

Pruebas de Aceptación:

- Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.
- Asociación bien asignada.

10	**BUSCAR EN EL SISTEMA**

Como usuario, quiero poder buscar y encontrar información en el sistema, para así acceder con mayor rapidez a lo deseado.

Estimación:	2	Riesgo:	Bajo
Prioridad:	100	Dependiente de:	

Pruebas de Aceptación:

- Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.
- Datos obtenidos correctamente.

11	**COMUNICACIÓN EN EL SISTEMA**
Como administrador y vendedor, quiero poder enviar correos a mis supervisores, para consultas o cualquier eventualidad que pudiera aparecer.	
Estimación: 6	Riesgo: Bajo
Prioridad: 400	Dependiente de:
Pruebas de Aceptación: • Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios. • Correo enviado y recibido satisfactoriamente.	

12	**SEGUIMIENTO DE CLIENTE Y PROSPECTO**
Como usuario, quiero poder registrar los pasos que se llevan a cabo con el prospecto, para volverse un cliente.	
Estimación: 4	Riesgo: Medio
Prioridad: 600	Dependiente de:
Pruebas de Aceptación: • Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios. • El prospecto queda registrado como cliente.	

13	**MARKETING**
Como usuario, quiero poder enviar correos de las promociones, nuevos productos y cambios en la empresa, para mantener interesado al cliente y poder hacer que compre más productos con nosotros.	
Estimación: 6	Riesgo: Bajo
Prioridad: 600	Dependiente de:

Pruebas de Aceptación:

- Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.
- Correo enviado y recibido satisfactoriamente.

14	**CONSULTAR REPORTES**		
Como usuario, quiero un sector donde pueda consultar los reportes determinados por la empresa, para llevar un control de cómo están los movimientos de la empresa.			
Estimación:	6	Riesgo:	Alto
Prioridad:	500	Dependiente de:	

Pruebas de Aceptación:

- Reportes son precisos y correctos.

2.5. TAREAS

A = Análisis, C = Codificación, D = Diseño, I = Investigación.

Tabla 7 Tareas

ID	Historia	Tarea	Tipo
1	1	Elaborar plantilla de inicio y formulario	D
2	1	Elaborar formulario login	D
3	1	Generar el código de mensajes, captura de datos y re direccionamiento formulario login.	C
4	1	Elaborar código de envió de correos re-utilizable.	C
5	1	Elaborar formulario recuperación.	D
6	1	Generar el código de mensajes, captura de datos y re direccionamiento formulario recuperación.	C
7	2	Elaborar plantilla interna y menú responsivo	D
8	2	Elaborar rutas en el sistema	C
9	3	Definir y generar tabla permission, rol y sus relaciones.	A
10	3	Elaboración de la plantilla de tablas.	D
11	3	Elaboración de la plantilla de administración de roles.	D

12	3	Generar código de organización de información dentro de la plantilla.	C
13	3	Elaboración del formulario de cambio.	D
14	3	Generar el código de mensajes, captura de datos y re direccionamiento.	C
15	4	Definir y generar tabla log y sus relaciones.	A
16	4	Elaborar el código de registrar Log re-utilizable.	C
17	5	Elaboración de la plantilla de perfil.	D
18	5	Generar código de organización de información dentro de la plantilla.	C
19	5	Elaboración de formulario de modificación de datos.	D
20	5	Generar el código de mensajes, captura de datos y re direccionamiento	C
21	6	Definir y generar tabla payment y sus relaciones.	A
22	6	Elaboración de plantilla de pagos pendientes.	D
23	6	Generar código de organización de información dentro de la plantilla.	C
24	6	Elaboración de plantilla de validación de pagos.	D
25	6	Generar el código de mensajes, captura de datos cambio de pago.	C
26	7	Definir y generar tabla business, contact y sus relaciones.	A
27	7	Elaboración de la plantilla de control de clientes y prospectos.	D
28	7	Generar código de organización de información dentro de la plantilla.	C
29	7	Elaboración de formularios de registro y modificación.	D
30	7	Generar el código de mensajes, captura de datos y re direccionamiento.	C
31	7	Elaboración de la plantilla de información.	D
32	7	Generar código de organización de información dentro de la plantilla.	C
33	7	Generar código de inserción de datos de cliente en formulario para actualización.	C
34	7	Elaboración de la plantilla de asociación de productos.	D
35	7	Generar el código de mensajes, captura de datos y re direccionamiento	C
36	8	Definir y generar tabla product y sus relaciones.	A
37	8	Elaboración de la plantilla de control de productos.	D
38	8	Generar código de organización de información dentro de la plantilla.	C
39	8	Elaboración de formulario de agregar producto.	D

40	8	Generar el código de mensajes, captura de datos y re direccionamiento.	C
41	9	Elaboración de la plantilla de administración.	D
42	9	Generar código de organización de información dentro de la plantilla.	C
43	9	Elaboración de formularios de registro y modificación.	D
44	9	Generar el código de mensajes, captura de datos y re direccionamiento.	C
45	9	Elaboración de la plantilla de información.	D
46	9	Generar código de organización de información dentro de la plantilla.	C
47	9	Generar código de inserción de datos de cliente en formulario para actualización.	C
48	10	Elaboración de la barra de búsqueda.	D
49	10	Generar código de búsqueda.	C
50	10	Generar código de obtención y organización de datos.	C
51	11	Elaboración de formulario de envío de correos.	D
52	11	Generar el código de mensajes, captura de datos y re direccionamiento.	C
53	12	Elaboración de formulario de envío de correos.	D
54	12	Generar el código de mensajes, captura de datos y re direccionamiento.	C
55	13	Definir y generar tabla record y sus relaciones.	A
56	13	Creación formulario de seguimiento de cliente y prospecto	D
57	13	Generar el código de mensajes, captura de datos y re direccionamiento.	C
58	14	Creación formulario de marketing.	D
59	14	Generar el código de mensajes, captura de datos y re direccionamiento.	C
60	15	Elaboración del diagrama circular con api de google para clientes.	I
61	15	Creación tabla de clientes asignados.	A
62	15	Generar el código de captura de datos.	C
63	15	Elaboración del diagrama circular con api de google para productos.	I
64	15	Creación tabla de productos vendidos.	D
65	15	Generar el código de captura de datos.	C

2.6. SPRINTS

2.6.1. Iteración 1

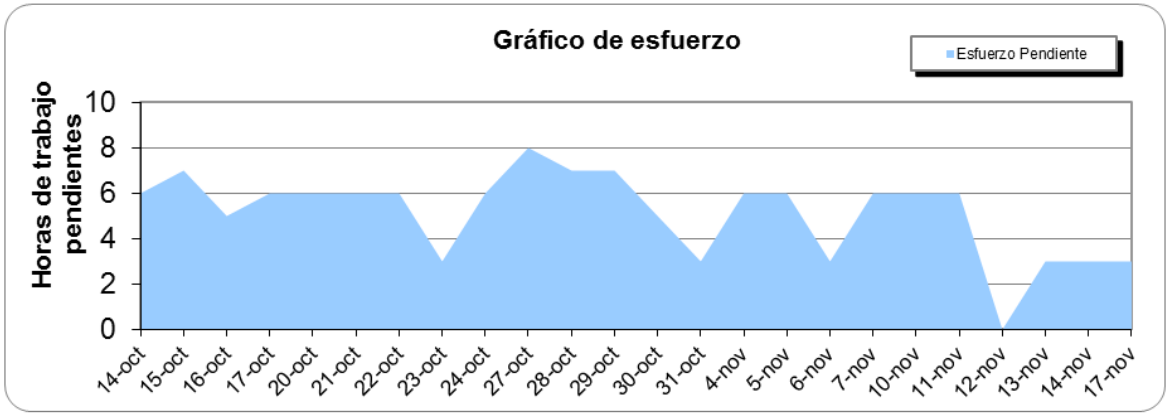
SPRINT BACKLOG o Lista de Historias a utilizar:

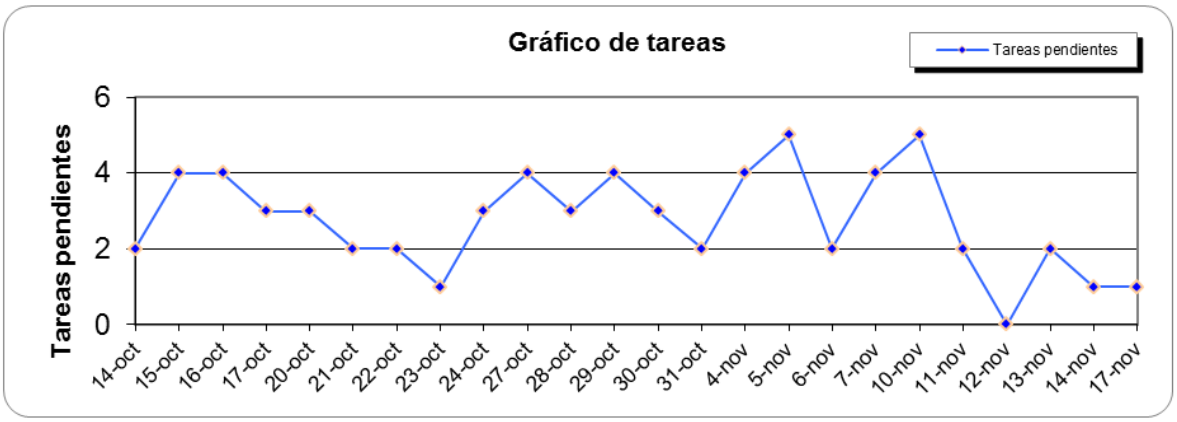
1	2	3	4	5	12		

Pila del Sprint:

Tabla 8 Pila Iteración 1

Tareas pendientes			1	3	4	3	3	2	2	1	3	4	3	4	7	2	4	5	2	4	5	2	0	2	1	1
Horas de trabajo pendientes			6	7	5	6	6	6	6	3	6	8	7	7	5	3	6	6	3	6	6	6	0	3	3	3
Tarea	**Estado**	**Responsable**	**Esfuerzo**																							
1	Terminada	Edward	3	1																						
2	Terminada	Edward		1																						
3	Terminada	Edward		2	1																					
4	Terminada	Manuel	3	3	2																					
5	Terminada	Edward			1																					
6	Terminada	Edward					3																			
7	Terminada	Edward			1	1		3	3	3	3	3														
8	Terminada	Edward										1														
9	Terminada	Manuel				3	1																			
10	Terminada	Edward				2																				
11	Terminada	Edward											3	1	1		1	1								
12	Terminada	Edward												2	2		2	1			1					
13	Terminada	Edward																1		1	1					
14	Terminada	Edward																		2	1	3				
15	Terminada	Manuel					2																			
16	Terminada	Manuel						3	3																	
17	Terminada	Manuel									1	2	3	2	2											
18	Terminada	Manuel									2	2	1	2												
19	Terminada	Manuel														1	1	1	1							
20	Terminada	Manuel														2	2	2	2							
36	Terminada	Manuel																				3				
55	Terminada	Manuel																		2	1					
56	Pendiente	Manuel																		1	2			1		
57	Pendiente	Manuel																						2	3	3





2.6.2. Iteración 2

SPRINT BACKLOG o Lista de Historias a utilizar:

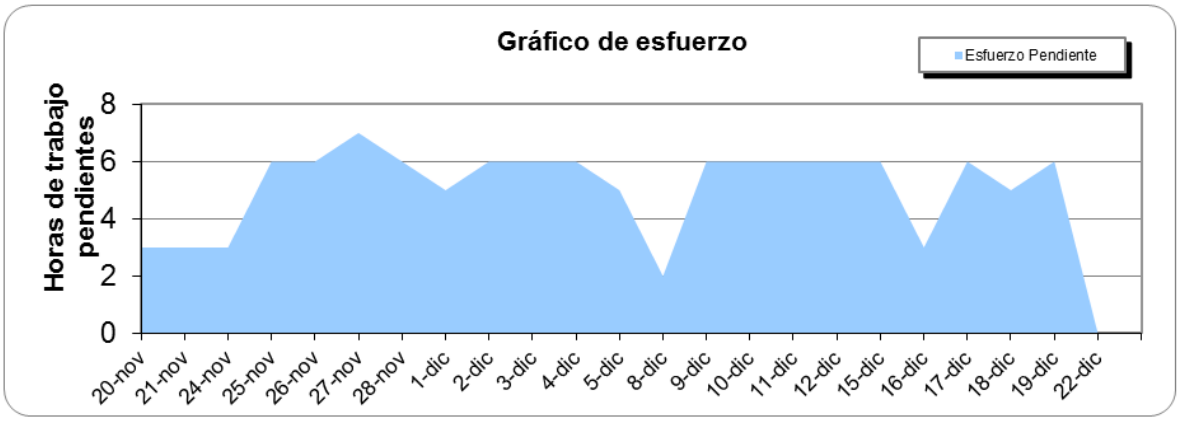
6	7	8	9	10	12		

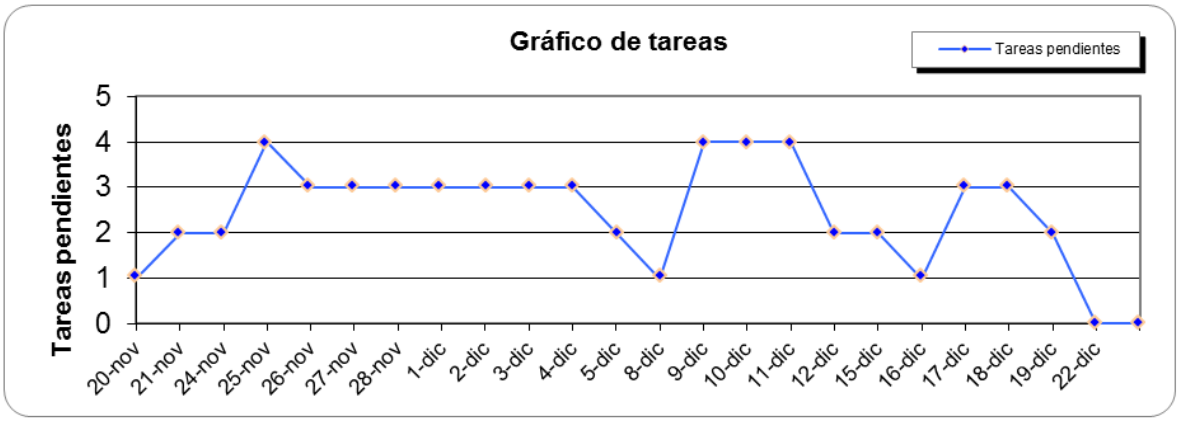
Pila del Sprint:

Tabla 9 Pila Iteración 2

Tareas pendientes			2	4	3	3	2	3	3	3	3	3	3	2	1	4	4	4	2	2	1	3	3	2
Horas de trabajo pendientes			6	5	4	6	6	7	6	5	6	6	6	5	2	6	6	6	6	6	3	6	5	6
Tarea	**Estado**	**Responsable**	**Esfuerzo**																					
21	Terminada	Manuel				2	1																	
22	Terminada	Manuel				1	2	2	1	1		2	1											
23	Terminada	Manuel						2	2	2	3	1				1	1	1	3					

24	Terminada	Manuel										2	2		2	2	2						
25	Terminada	Manuel																					
26	Terminada	Manuel																3					
27	Terminada	Edward	3	2	2	1																	
28	Terminada	Edward					3	3	3														
29	Terminada	Edward		1	1	2				2	2												
30	Terminada	Edward									1	3	3	3									
31	Terminada	Edward													2	2	1	1					
32	Terminada	Edward																	3	3	3	2	
33	Pendiente	Edward																				1	2	3
34	Terminada	Edward													1	2	2						
35	Pendiente	Edward																					
36	Terminada	Manuel																				3	
37	Pendiente	Manuel																					1
38	Pendiente	Manuel																					2	3
39	Pendiente	Manuel																					
40	Pendiente	Manuel																					
41	Pendiente	Edward																					
42	Pendiente	Edward																					
43	Pendiente	Edward																					
44	Pendiente	Edward																					
45	Pendiente	Edward																					
46	Pendiente	Edward																					
47	Pendiente	Edward																					
48	Pendiente	Edward																					
49	Pendiente	Edward																					
50	Pendiente	Edward																					
56	Terminada	Manuel		1	1																		
57	Terminada	Manuel	3	1																			





2.6.3. Iteración 3

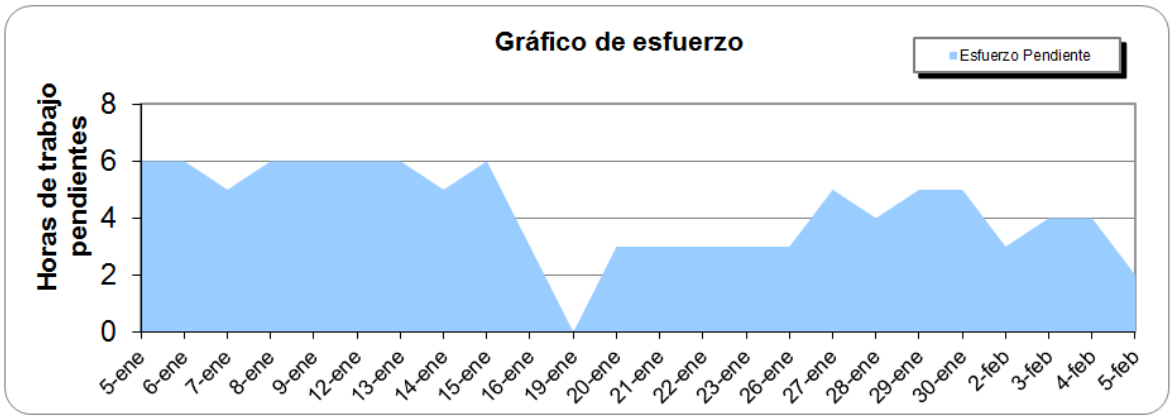
SPRINT BACKLOG o Lista de Historias a utilizar:

7	8	9	10				

Pila del Sprint:

Tabla 10 Pila Iteración 3

Tareas pendientes			2	3	2	3	3	3	3	3	3	1	0	1	1	1	1	2	3	3	3	4	1	3	2	1
Horas de trabajo pendientes			6	6	5	6	6	6	6	5	6	3	0	3	3	3	3	3	5	4	5	5	3	4	4	2
Tarea	**Estado**	**Responsable**	**Esfuerzo**																							
33	Terminada	Edward	3																							
35	Terminada	Edward		3	3	3																				
37	Terminada	Manuel	1	2	2																					
38	Terminada	Manuel	2	1																						
39	Terminada	Manuel				1	1			2	3										2	1		1		
40	Terminada	Manuel				2	2	3	3										2	2		1		1		
41	Terminada	Edward					3	2	2	2																
42	Terminada	Edward						1			1															
43	Terminada	Edward							1	1	2	3														
44	Terminada	Edward												3	3	3	3									
45	Terminada	Edward																2	1	1						
46	Terminada	Edward																			2	2		2	3	
47	Pendiente	Edward																			1	1	3		1	2
48	Terminada	Edward																1	2	1						
49	Pendiente	Edward																								
50	Pendiente	Edward																								





2.6.4. Iteración 4

SPRINT BACKLOG o Lista de Historias a utilizar:

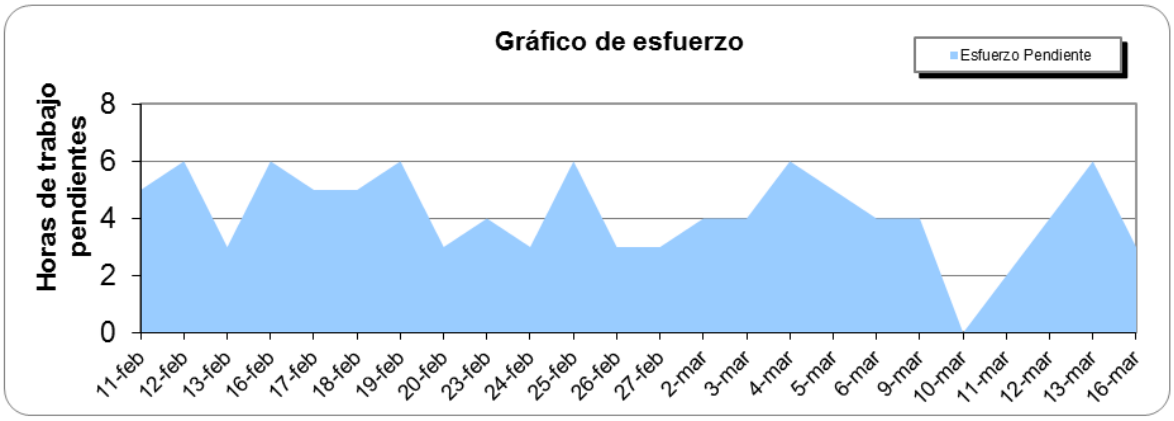
9	10	11	12	14			

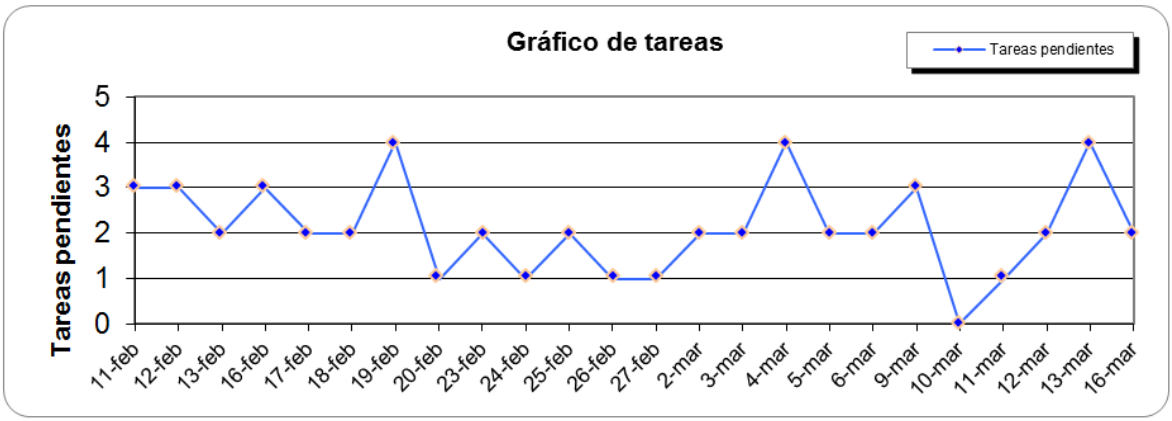
Pila del Sprint:

Tabla 11 Pila Iteración 4

Tareas pendientes			2	3	2	3	2	2	4	1	2	1	2	1	1	2	2	4	2	2	3	0	1	2	4	2
Horas de trabajo pendientes			5	6	3	6	5	5	6	3	4	3	6	3	3	4	4	6	5	4	4	0	2	4	6	3
Tarea	**Estado**	**Responsable**	**Esfuerzo**																							
47	Terminada	Edward	2																							
49	Terminada	Edward		2	2	2	2		1																	
50	Terminada	Edward		1	1	1		2	2	3	2															
51	Terminada	Manuel	1				3	3	2																	

52	Terminada	Manuel	2	3		3			1																
53	Terminada	Manuel									2							2	3	2	1			2	2	
54	Terminada	Manuel										3	3			2	2	1			1				1	
58	Pendiente	Manuel																								
59	Pendiente	Manuel																								
60	Terminada	Edward											3	3	3	2	2	2							1	2
61	Terminada	Edward																1	2	2	2					
62	Terminada	Edward																					2	2	2	1
63	Pendiente	Edward																								
64	Pendiente	Edward																								
65	Pendiente	Edward																								





2.6.5. Iteración 5

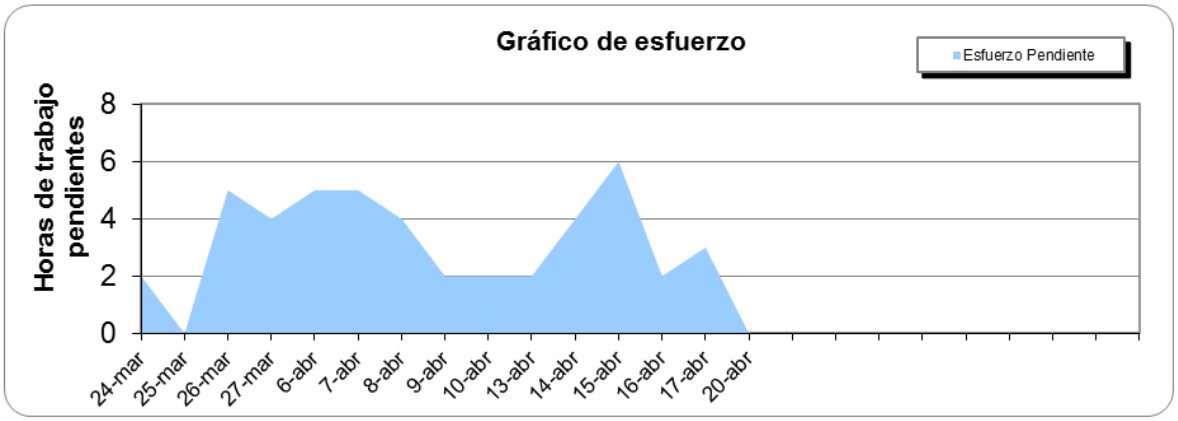
SPRINT BACKLOG o Lista de Historias a utilizar:

12	14						

Pila del Sprint:

Tabla 12 Pila Iteración 5

Tareas pendientes			1	0	2	2	3	2	1	1	1	2	1	2	1	1
Horas de trabajo pendientes			2	0	5	4	5	5	4	2	2	2	4	6	2	3
Tarea	**Estado**	**Responsable**	**Esfuerzo**													
58	Terminada	Manuel	1					2	2				2			1
59	Terminada	Manuel	1		3	2	2					1		3	2	2
63	Terminada	Edward			2	2				2	2					
64	Terminada	Edward					2	2								
65	Terminada	Edward					1	1	2			1	2	3		





Observaciones del sprint:

Terminación de tareas, y pruebas de aceptación.

2.6.6. Iteración 6

Observaciones del sprint:

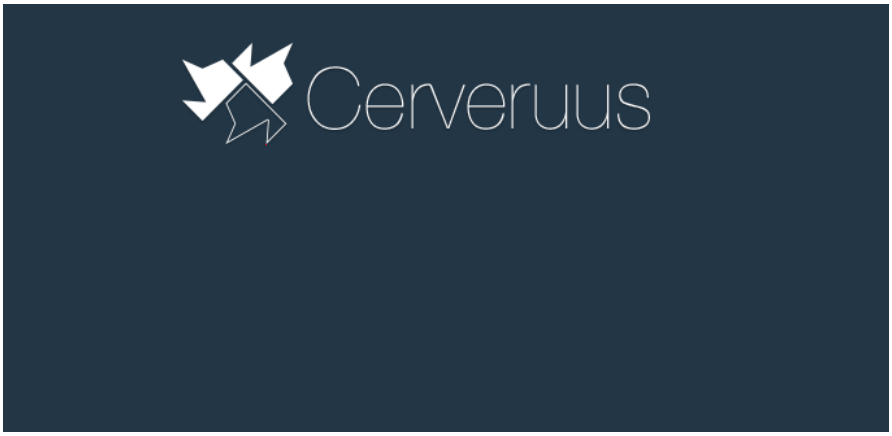
Sprint dedicado a la resolución de problemas presentados en el sistema y a las pruebas de aceptación de cada historia de usuario.

3. FASE DE DESARROLLO

- **INGRESAR AL SISTEMA**

DESARROLLO DE TAREAS:

- ✓ **Elaborar plantilla de inicio y formulario.**



Grafica 2 Plantilla de entrada



Grafica 3 Componentes Formulario login

Parte de código css con mediaQuery (@media):

```
input[type=submit] {
    font-family: 'Questrial', sans-serif;
    font-size: 16px;
    width: 100%;
    padding-right:10px;
    padding-left: 10px;
    padding-bottom: 5px;
    padding-top: 5px ;
    border: 3px solid #069ec7;
    color: #233645;
    background: #ffffff;
    font-weight: 600;
    margin-left: 7px;
}

input[type=password] {
    border: 2px solid #069ec7;
    font-family: 'Questrial', sans-serif;
    font-size: 16px;
    width: 100%;
    margin-bottom: 8px;
    padding: 5px;
    background:#07b4e3;
```

```
   color: #ffffff;
   background:none repeat scroll 0 0 rgba(35, 54, 69, 1);
}

@media all and (min-width: 0px) and (max-width: 800px){

   #ima img{
      display: block;
      margin: auto;
      width: 100%;
   }

   .container{
      width: 70%;
   }

   .margen{
      width: 100%;
   }

   .enviar{
      width: 100%;
   }
}
```

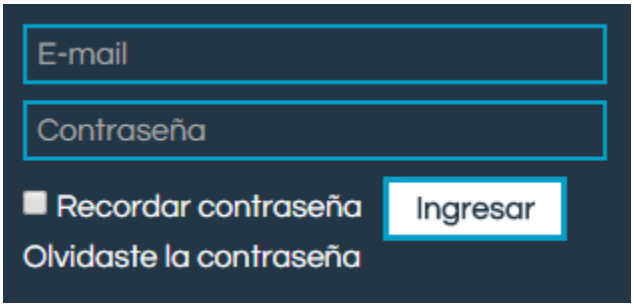
- ✓ **Definir y generar tabla User.**



Grafica 4 Tabla User

USER			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
	Address	varchar(255)	Dirección de residencia actual
	Email	varchar(255)	Correo electrónico
	Last_name	varchar(255)	Apellidos
	Name	varchar(255)	Nombres
	Password	varchar(255)	Contraseña de sesión
	Phone	int(11)	Número telefónico
	Photo	varchar(255)	Ruta de la foto de perfil
	Remember_token	varchar(255)	Recuerda el email y password del login
	Restore_password	varchar(255)	Sirve para que el usuario pueda ingresar a una ruta específica para restaurar su contraseña
	User_name	varchar(255)	Nombre de usuario
FK	Role_id	int(11)	Identificación del rol
FK	Manager	int(11)	Usuario a cargo

- ✓ **Elaborar formulario login.**



Grafica 5 Formulario login

- ✓ **Generar el código de mensajes, captura de datos y re direccionamiento formulario login.**

```
public function login()
   {
      $data = Input::only('email', 'password', 'remember');
      $credentials = ['email' => $data['email'], 'password' => $data['password']];

      if (Auth::attempt($credentials, $data['remember'])) {
         return Redirect::route('home');
      }

      return Redirect::to('login')
         ->with('mensaje_error', 'Tu correo o contraseña son incorrectos')
         ->withInput();
}
```

La clase Auth permite la autentificación de los usuarios y al pasarle los atributos a su método attempt conecta con la base de datos y verifica la contraseña ya encriptada y evalúa la legitimidad de los datos.

- ✓ **Generar el código de envió de correo.**

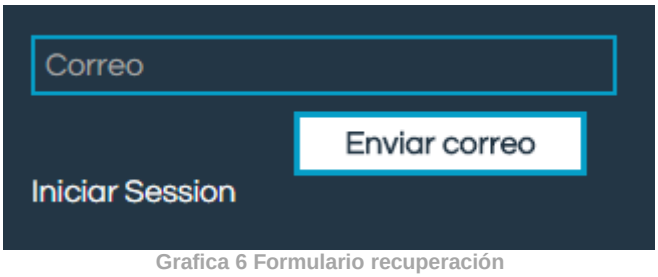
```
public function sendRestore()
   {
      $userRepo= new UserRepo();
      $validator=$userRepo->passwordRestart(Input::get('email'));
      $data=['link'=>$validator['link']];
      if($validator['return'])
      {
         Mail::send('emails.password', $data, function ($message) {
            $message->to(Input::get('email'), 'cerverus')->subject('correo de restauracion de su password');
         });
      }else{
         return Redirect::to('restaurar')->with('mensaje_error','tu correo no se encuentra registrado')->withInput();
      }
      new LogRepo(
         [
            'responsible'=> $validator['username'],
            'responsible_id'=> $validator['id'],
            'action' => 'ha solicitado restaurar la contraseña',
         ]
      );
      return Redirect::to('login')->with('mensaje_error','El correo fue enviado con exito')->withInput();
}
```

Laravel proporción una clase llamada Mail la cual es la encargada de conexiones remotas con servicios de mensajería, validación y estructuración del correo a enviar. El código empleado es:

```
        'mailgun' => array(
                'domain' => '************@****',
                'secret' => '*********',
        ),
        'mandrill' => array(
                'secret' => '************@****',
        ),
        'stripe' => array(
                'model'  => 'User',
                'secret' => '',
        ),
```

Ya configurado tenemos acceso a nuestro servicio de mensajería (ej. Gmail.com, Hotmail.com) y podremos llamar a los métodos de Mail para la configuración y envió de correos. Con esto expuesto no se hará más referencia a la clase Mail en el desarrollo del proyecto.

- ✓ **Elaborar formulario recuperación.**



Grafica 6 Formulario recuperación

- ✓ **Generar el código de mensajes, captura de datos y re direccionamiento formulario recuperación.**

```
public function restorePassword($restore_password)
   {
      $userRepo = new UserRepo();
      $validator = $userRepo->validatorUser($restore_password);
      if($validator)
      {
         $user=['id'=>$restore_password];
         return View::make('front.restorePassword',compact('user'));
      }
      return Redirect::route('home');
}
```

PRUEBAS DE ACEPTACIÓN:

- ✓ **Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.**



Grafica 7 Formularios excepciones

- **NAVEGACION EN EL SISTEMA**

DESARROLLO DE TAREAS:

- ✓ **Elaborar plantilla interna y menú responsivo.**



Grafica 8 Plantilla interna y menú responsivo

- ✓ **Elaborar rutas en el sistema.**

```
<?php
//SON LAS RUTAS EN LAS QUE SE LES ENVIA UN CORREO CON EL LINK PARA RESTAURAR LA CONTRASEÑA
route::get('restaurar', ['as' => 'restore', 'uses' => 'AuthController@showRestore']);
route::post('restaurar', ['as' => 'restore', 'uses' => 'AuthController@sendRestore']);
route::get('restaurarContraseña/{id}', ['as' => 'restorePassword', 'uses' => 'AuthController@restorePassword']);
route::post('restaurarContraseña/{id}', ['as' => 'restorePassword', 'uses' => 'AuthController@changePassword']);
//INICIO DE SESSION
route::get('login', ['as' => 'login', 'uses' => 'AuthController@signUp']);
route::post('login', ['as' => 'login', 'uses' => 'AuthController@login']);
//Aqui va las rutas que puede acceder cuando esta logeado
Route::group(array('before' => 'auth'), function()
{
    Route::get('/', function()
    {
        $permiso =new Proceso();
        $total=$permiso->filtrarPermisos();
        return View::make('front.home',compact('total'));
    });
    route::get('Inicio', ['as' => 'home', 'uses' => 'HomeController@index']);
    route::get('perfil', ['as' => 'profile', 'uses' => 'UserController@showProfile']);
    route::post('actualizardatos', ['as' => 'updateProfile', 'uses' => 'UserController@updateProfile']);
    route::post('cambiarContraseña', ['as' => 'changePassword', 'uses' => 'UserController@changePassword']);
    Route::group(array('before'=>'business'),function() {
        route::get('admin/Empresas', ['as' => 'business', 'uses' => 'BusinessController@showBusiness']);
        route::post('admin/Empresas', ['as' => 'business', 'uses' => 'BusinessController@saveBusiness']);
        route::get('admin/actualizarEmpresa/{id}', ['as' => 'updateBusiness', 'uses' => 'BusinessController@showUpdateBusiness']);
        route::post('admin/actualizarEmpresa/{id}', ['as' => 'updateBusiness', 'uses' => 'BusinessController@updateBusiness']);
        route::get('admin/verEmpresa/{id}', ['as' => 'seeBusiness', 'uses' => 'BusinessController@showSeeBusiness']);
        route::post('admin/verEmpresa/{id}', ['as' => 'seeBusiness', 'uses' => 'BusinessController@saveStates']);
        route::get('admin/pagos/{id}',['as'=>'paymentBusiness','uses'=>'BusinessController@showPayment']);
        route::post('admin/pagos/{id}',['as'=>'paymentBusiness','uses'=>'BusinessController@updatePayment']);
    });
    Route::group(array('before'=>'contacts'),function(){
        route::get('admin/contactos', ['as' => 'contacts', 'uses' => 'ContactController@contact']);
        route::post('admin/contactos', ['as' => 'contacts', 'uses' => 'ContactController@createContact']);
        route::post('admin/cargos', ['as' => 'charges', 'uses' => 'ContactController@createCharges']);
        route::get('admin/eliminar/{id}', ['as' => 'deleteCharges', 'uses' => 'ContactController@deleteCharges']);
        route::get('admin/actualizar/{id}', ['as' => 'updateContacts', 'uses' => 'ContactController@showUpdateContact']);
        route::post('admin/actualizar/{id}', ['as' => 'updateContacts', 'uses' => 'ContactController@updateContact']);
    });
    Route::group(array('before'=>'administrator'),function(){
        route::get('admin/cuentas', ['as' => 'administrator', 'uses' => 'UserController@showUsers']);
    });
    Route::group(array('before'=>'roles'),function(){
        route::get('admin/roles', ['as' => 'roles', 'uses' => 'PermissionController@show']);
        route::get('admin/role/{id}', ['as' => 'showPermissions', 'uses' => 'PermissionController@showPermissions']);
        route::post('admin/actualizarPermisos/{id}', ['as' => 'updatePermissions', 'uses' =>
'PermissionController@updatePermissions']);
    });
    Route::group(array('before'=>'createUser'),function(){
        route::get('admin/crearUsuario', ['as' => 'createUser', 'uses' => 'UserController@createUser']);
        route::post('admin/crearUsuario', ['as' => 'createUser', 'uses' => 'UserController@saveUser']);
    });
    Route::group(array('before'=>'showUser'),function(){
        route::get('admin/ver/{id}', ['as' => 'showUser', 'uses' => 'UserController@showUser']);
        route::post('admin/ver/{id}', ['as' => 'showUser', 'uses' => 'UserController@showUser']);
    });
    Route::group(array('before'=>'updateUser'),function(){
        route::get('admin/actualizarUsuario/{id}', ['as' => 'updateUser', 'uses' => 'UserController@showUpdateUser']);
        route::post('admin/actualizarUsuario/{id}', ['as' => 'updateUser', 'uses' => 'UserController@updateUser']);
    });
    Route::group(array('before'=>'deleteUser'),function(){
        route::get('admin/eliminarUsuario/{id}', ['as' => 'deleteUser', 'uses' => 'UserController@deleteUser']);
    });
    Route::get('logout', 'AuthController@logOut');
});
```

PRUEBAS DE ACEPTACIÓN:

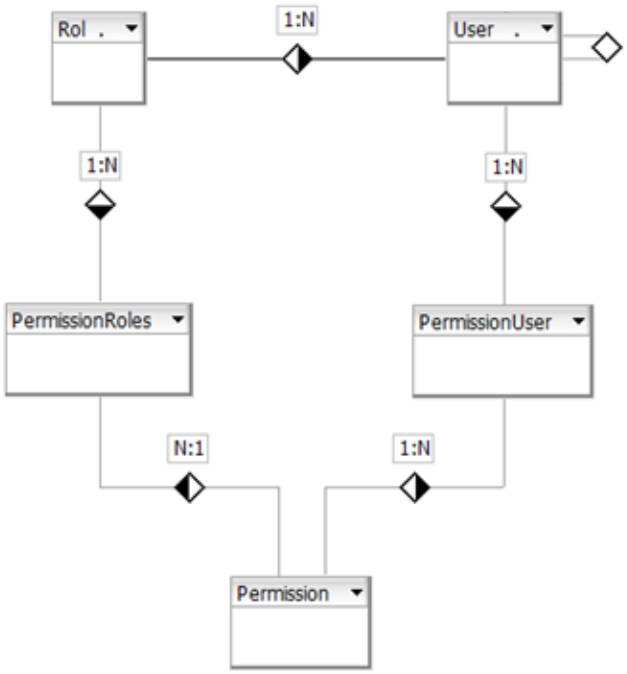
- ✓ **Se re direcciona correctamente a todas las áreas en el sistema.**

Todas las redirecciones son correctas y en rutan el sistema adecuadamente.

- **DAR PERMISOS DE ACCESO**

DESARROLLO DE TAREAS:

- ✓ **Definir y generar tabla permission, rol y sus relaciones.**



Grafica 9 Tabla permission, rol y sus relaciones

ROL			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
	Name	varchar(255)	Nombre del role

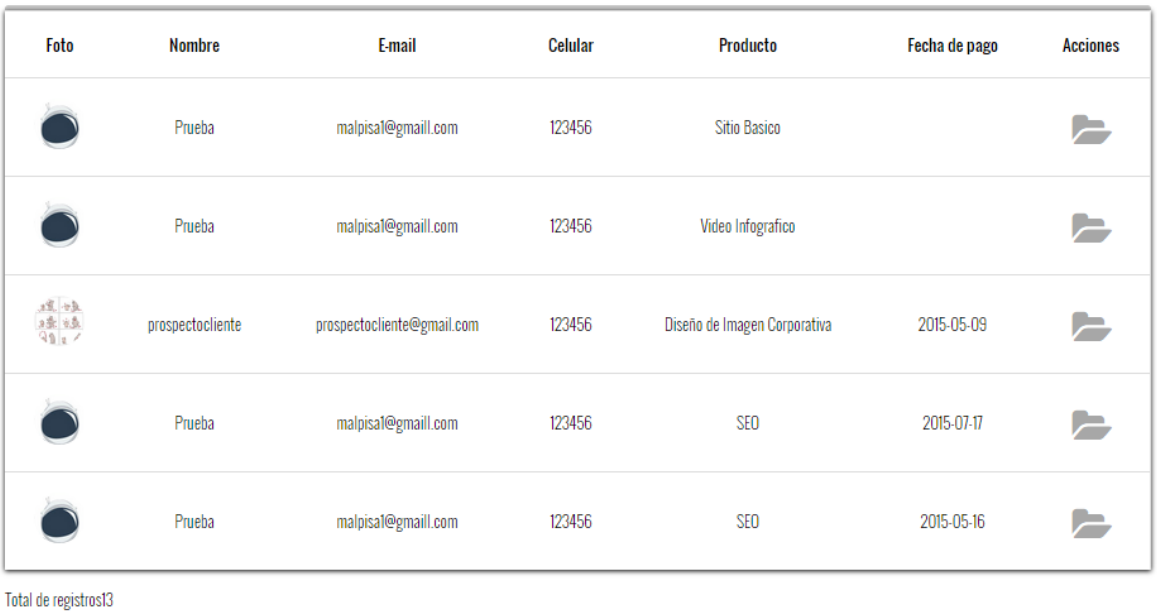
PERMISSION			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
FK	Name	varchar(255)	Nombre

PERMISSIONROL			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
FK	Rol_id	int(11)	Identificación del rol
FK	Permission_id	int(11)	Identificación del permiso

PERMISSIONUSER			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
	Permission_no	int(11)	Número del permiso
FK	User_id	int(11)	Identificación del usuario
FK	Permission_id	int(11)	Identificación del permiso

- ✓ **Elaboración de la plantilla de tablas.**

La plantilla presentada contiene datos de prueba para la visualización clara de los elementos de la misma.

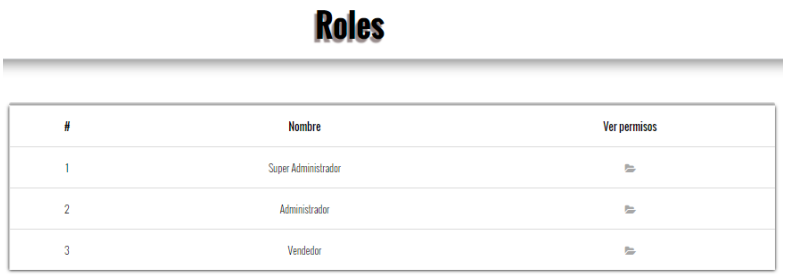


Grafica 10 Plantilla tabla

Código css:

```
.tableContent{
   box-shadow: 0 1px 5px 0;
   overflow: auto;
   text-align: center;
}
table{
   background-color: #ffffff;
   box-shadow: 0 1px 3px 0 rgba(0, 0, 0, 0.12), 0 1px 2px 0 rgba(0, 0, 0, 0.24);
   margin: 0 auto;
   overflow: auto;
   text-align: center;
   width: 100%;
}
th,td {
   border-top: 1px solid #ddd;
   padding: 16px;
}
th{
   font-weight: 500;
}
td{
   font-weight: 100;
}
.tableContent a{
   color: rgba(0, 0, 0, 0.37);
   font-size: 18px;
}
.tableContent a:hover{
   color: #07b4e3;
}
.tableContent img{
   border-radius: 50%;
   height: 40px;
   width: 40px !important;
}
```

- ✓ **Elaboración de la plantilla de administración de roles.**



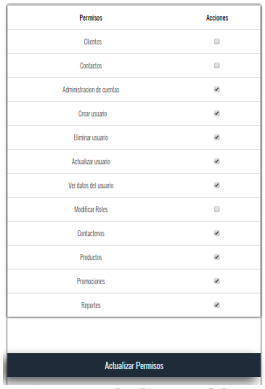
Grafica 11 Roles

- ✓ **Generar código de organización de información dentro de la plantilla.**

```
<div class="wrapperContent">
   <div class="tableContent">
      <table>
         <thead>
         <tr>
            <th>#</th>
            <th>Nombre</th>
            <th>Ver permisos</th>
         </tr>
```

```
            </thead>
            <tbody>
            @foreach($roles as $role)
               <tr>
                  <td>{{$role->id}}</td>
                  <td>{{$role->name}}</td>
                  <td><a href="{{route('showPermissions',$role->id)}}" class="icon-folder-open "></a></td>
               </tr>
            @endforeach
            </tbody>
         </table>
      </div>
   </div>
```

✓ Elaboración del formulario de cambio.



Grafica 12 Formulario cambio permiso

✓ Generar el código de mensajes, captura de datos y re direccionamiento.

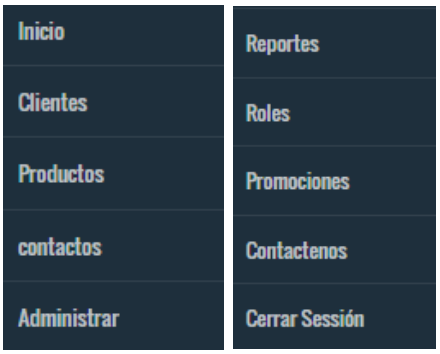
```
public function updatePermissions($id)
{
      if($id>1)
      {
         PermissionRole::where('role_id', '=', $id)->delete();
         $permission=Permission::all();

         for($i=1;$i<=count($permission);$i++)
         {
            if(Input::get(''.$i.''))
            {
               $permissionRole = new PermissionRole(['role_id' => $id, 'permission_id' => Input::get('' . $i . '')]);
               $permissionRole->save();
            }
         }
         new LogRepo(
            [
               'responsible'=> Auth::user()->user_name,
               'responsible_id'=> Auth::user()->id,
               'action' => 'ha actualizado permisos de role '
            ]
         );
         return Redirect::to('/admin/role/'.$id)->with('message','Se actualizaron los permisos correctamente');
      }
      return Redirect::to('/admin/role/'.$id)->with('message_error','Al Super Administrador no se le puede modificar los
permisos');
}
```

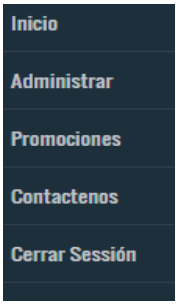
PRUEBAS DE ACEPTACIÓN:

- ✓ **Intentar acceder a un sector no permitido.**

A todo rol se le ha asignado permisos específicos y aquellos que no tengan el permiso se les oculta y/o bloquea determinada acción.



Grafica 13 Permisos completos



Grafica 14 Permisos parciales

Todos los permisos han sido comprobados y se verificado que ningún usuario puede hacer una acción no permitida.

- **REGISTRAR ACTIVIDADES (LOGS)**

DESARROLLO DE TAREAS:

- ✓ **Definir y generar tabla log y sus relaciones.**



Grafica 15 Tabla log y sus relaciones

LOG			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
	Action	varchar(255)	Acción realizada por usuario
	Affected_entity	varchar(255)	Qué modelo o tabla afecto
FK	Affected_entity_id	int(11)	Id de la tabla
	Responsible	varchar(255)	Responsable de la acción
FK	Responsible_id	int(11)	Id del responsable de la acción

- ✓ **Elaborar el código de registrar Log re-utilizable.**

```
class LogRepo extends BaseRepo{
   private $_data;

   public function __construct($data){
      $this->model = $this->getModel();
      $this->_data= $data;
      $this->saveLog();
   }

   protected function getModel(){
      return new Log();
   }


   public function saveLog()
   {
      $this->model->fill($this->_data) ;
      $this->model->save();
   }
}
```

En cualquier sector donde se realice un cambio se agrega el siguiente código:

```
new LogRepo([
            'responsible'=> $user['username'],
            'responsible_id'=> $user['id'],
            'action' => 'Accion Realizada',
]);
```

Con esto el Log queda guardado en la base de datos.

PRUEBAS DE ACEPTACIÓN:

- ✓ **Obtener los datos de una actividad realizada.**

```
id | responsible | responsible_id | affected_entity | action | created_at | updated_at | affected_entity_id |

| 1 | drawde | 1 |   | ha restaurado la contraseña | 2015-04-05 17:28:13 | 2015-04-05 17:28:13 | 0 |
| 2 | drawde | 1 | admin | ha creado una cuenta | 2015-04-05 17:37:07 | 2015-04-05 17:37:07 | 0 |
    .
    .
| 63 | malpisa | 1 |    | ha creado fechas de pago de una empresa  | 2015-04-11 18:35:03 | 2015-04-11
18:35:03 | 0 |
| 64 | malpisa | 1 |    | ha modificado los pagos de la empresa    | 2015-04-11 18:35:11 | 2015-04-11
18:35:11 | 0 |
```

Datos llegan correctamente a la BD.

- **PERFIL**

DESARROLLO DE TAREAS:

✓ Elaboración de la platilla de formularios internos.



Grafica 16 Plantilla Formulario

Código css:

```
.formContent{
    background: white;
    box-shadow: 0 1px 5px 0;
    border-radius: 5px;
    margin-bottom: 20px;
    padding: 20px;
}

form{
    border-radius: 15px;
    font-weight: 100;
    margin: 16px auto;
    width: 75%;
}


form div label{
    display: block;
    padding: 20px 10%;
    width: 100%;
}


form div input, form div select{
    background: white;
    border:none;
    border-radius: 15px;
    box-shadow: -7px 10px 20px rgba(0, 0, 0, 0.83);
    height: 20px;
    margin: 0px 10%;
    padding-left: 20px;
    width: 80%;
}
form div select{
    width: 83%;
}
```

```
input[type=submit]{
   background: rgba(29, 44, 56, 0.99);
   border: 1px solid rgba(29, 44, 56, 0.99);
   color: white;
   font-size: 18px;
   font-weight: 100;
   height: 4rem;
   margin: 3.4rem 0;
   width: 100%;
   box-shadow: -10px 10px 32px rgba(0, 0, 0, 0.65);
}

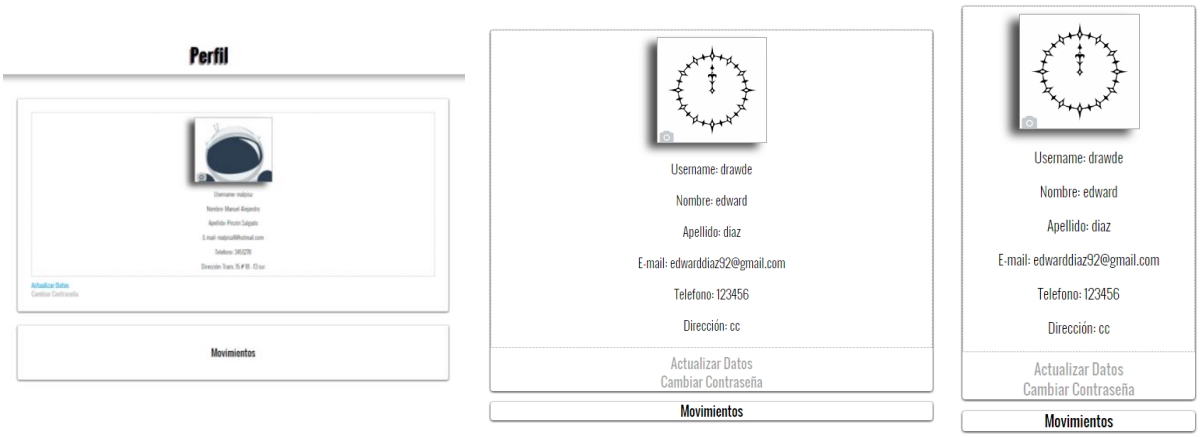
 textarea{
   background: white;
   border-radius: 15px;
   box-shadow: -7px 10px 20px rgba(0, 0, 0, 0.83);
   border:none;
   height: 200px;
   margin: 0px 10%;
   padding-left: 20px;
   width: 80%;
}

.formErrors{
   color: red;
   margin: 10px 10% 0px;
}

.formPayments label{
   text-align: center;
   padding: 20px 0;
}

.formPayments p{
   text-align: center;
   font-weight: 600;
}
```

✓ Elaboración de la plantilla de perfil.



Grafica 17 Perfil de usuario

Parte de código css:

```
.wrapperBody, .wrapperBody1{
   display: inline-block;
   vertical-align: top;
}

.wrapperBody{
   margin-right: -4px;
   width: 220px;
}

.wrapperBody1{

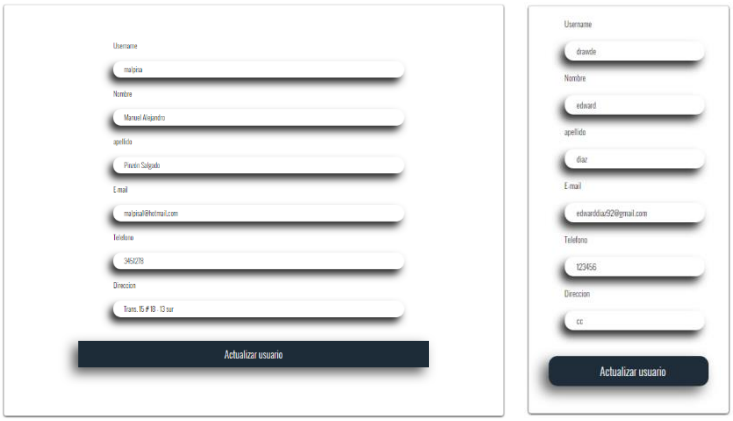
   width: 82.8125%;
   margin-right: -4px;
}

@media all and (min-width:1025px) and (max-width: 1375px){
   .wrapperBody1{
      width: 75.8125%;
   }
}
```

- ✓ **Generar código de organización de información dentro de la plantilla.**

```
<section>
        <figure>
          {{ HTML::image('user/'.$user->photo,'',array('id'=>'')) }}
        </figure>
        <p>Username: {{$user->user_name}}</p>
        <p>Nombre: {{$user->name}}</p>
        <p>Apellido: {{$user->last_name}}</p>
        <p>E-mail:  {{$user->email}}</p>
        <p>Telefono: {{$user->phone}}</p>
        <p>Dirección: {{$user->address}}</p>
        <p>
          @if($user->role_id==3)
                    A cargo de :{{$manager->user_name}}
          @endif
        </p>
</section>
```

- ✓ **Elaboración de formulario de modificación de datos.**



Grafica 18 Formulario modificación datos personales

- ✓ **Generar el código de mensajes, captura de datos y re direccionamiento.**

```
public function updateProfile()
{
    $user=User::find(Auth::user()->id);
    $updateProfile=new UpdateProfileManager($user,Input::all());
    $updateValidator=$updateProfile->isValid();
    if($updateValidator)
    {
        return Redirect::route('profile')->with('upload','1')->withErrors($updateValidator)->withInput();
    }

    $updateUser=$updateProfile->updateUser();
    if($updateUser)
    {
        new LogRepo(
            [
                'responsible'=> Auth::user()->user_name,
                'responsible_id'=> Auth::user()->id,
                'action' => 'ha actualizado sus datos ',
                'affected_entity'=> '',
                'affected_entity_id'=> '',
            ]
        );
        return Redirect::route('profile')->with('message','sus datos fueron actualizados exitosamente');
    }
    return Redirect::route('profile')->with('message_error','sus datos no puedieron se actualizados');
}
```

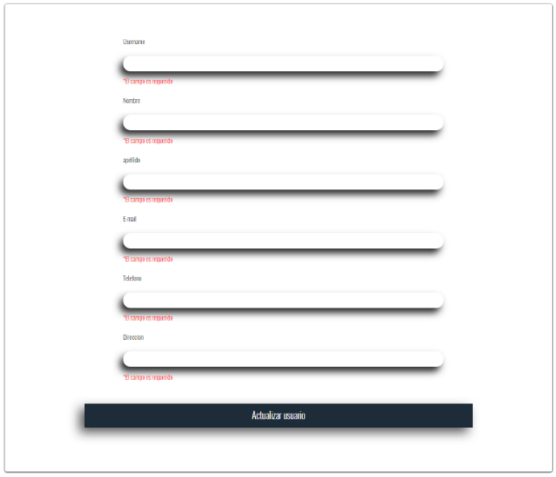
PRUEBAS DE ACEPTACIÓN:

- ✓ **Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.**

Todo input en el formulario se crea según su tipo, máximo, mínimo, obligación ej.

```
<input type="tipo" max-length="numero" required>Text</input>
```

Y gracias a la validaciones HTML y al código `{{$errors->first('mensaje')}}` podemos poner el mensaje de error para que aparezca debajo de la caja como muestra la Grafica 19.



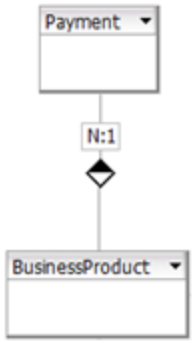
Grafica 19 Formulario excepciones

Con esto en las siguientes pruebas de aceptación de este tipo solo es necesario obtener la confirmación de aparición de los mensajes de error.

- **PAGOS PENDIENTES**

DESARROLLO DE TAREAS:

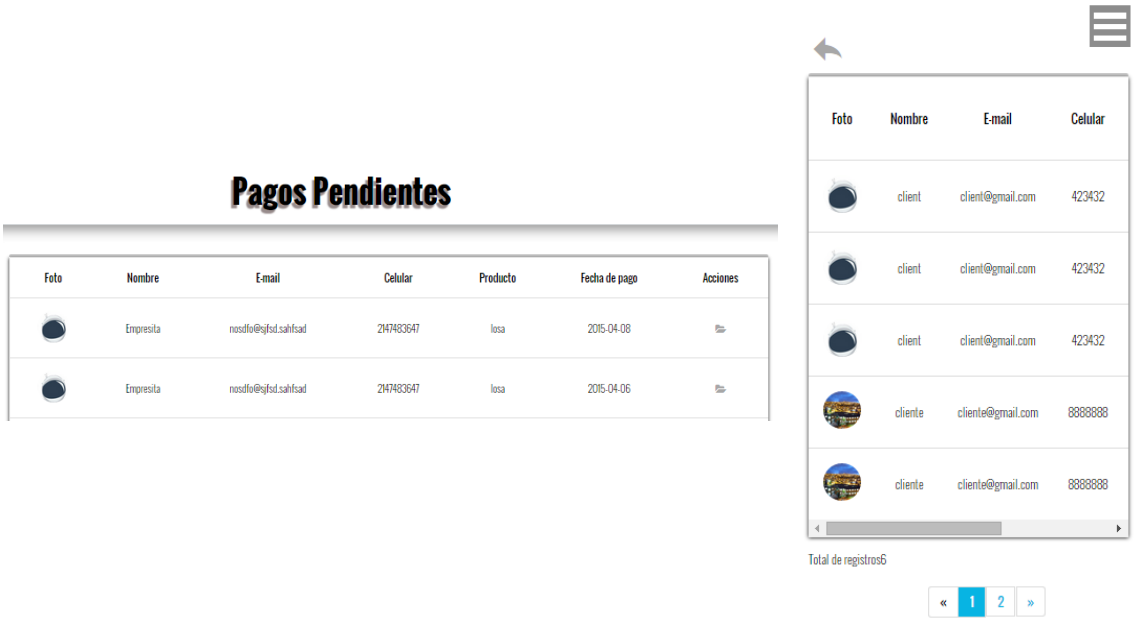
✓ **Definir y generar tabla payment y sus relaciones.**



Grafica 20 Tabla payment y sus relaciones

PAYMENT			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
	Name	varchar(255)	Nombre del producto comprado
	Payment	varchar(255)	Fecha del pago
	Type	varchar(255)	Determina a cuantas cuotas se realiza
	Validator	int(11)	Verifica si el pago se ha realizado
FK	BussinesProduct_id	varchar(255)	Identificación de la tabla BussinesProduct

✓ **Elaboración de plantilla de pagos pendientes.**



Grafica 21 Plantilla ingreso

✓ **Generar código de organización de información dentro de la plantilla.**

```
<div class="wrapperContent">
   <section class="tableContent">
      <table class=" ">
         <thead>
            <tr>
               <th>Foto</th>
               <th>Nombre</th>
               <th>E-mail</th>
               <th>Celular</th>
               <th>Producto</th>
               <th>Fecha de pago</th>
               <th>Acciones</th>
            </tr>
         </thead>
         <tbody>
         @foreach($payments as $payment)
            @if($payment->type>0)
               <tr>
                  @foreach($businessProducts as $businessProduct)
                     @if($payment->businessProduct_id==$businessProduct->id)
                        @foreach($business as $client)
                           @if($businessProduct->business_id==$client->id)
                              <td>{{HTML::image('business/'.$client->photo,'',array('style'=>'width:100px;'))}} </td>
                              <td> {{$client->name}}</td>
                              <td>{{$client->email}}</td>
                              <td>{{$client->mobile_phone}}</td>
                           @endif
                        @endforeach
                        @foreach($products as $product)
                           @if($businessProduct->product_id==$product->id)
                              <td> {{$product->name}}</td>
                           @endif
                        @endforeach
                     @endif
                  @endforeach
```

```
                <td>{{$payment->payment}}</td>
                <td><a href="{{route('paymentBusiness',$payment->businessProduct_id)}}">Validar el pago</a></td>
</tr>
            @endif
        @endforeach
        </tbody>
    </table>
  </section>
</div>
```

- ✓ **Elaboración de plantilla de validación de pagos.**



Grafica 22 Validación de pagos

- ✓ **Generar el código de mensajes, captura de datos cambio de pago.**

PRUEBAS DE ACEPTACIÓN:

- ✓ **Los clientes están a mi cargo.**

Todos los clientes entregados están a cargo del usuario.

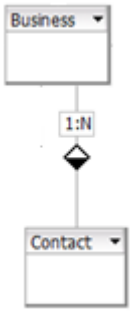
- ✓ **La validación es correcta.**

Toda validación es correcta y confirmada.

- **CONTROLAR CLIENTES Y PROSPECTOS**

DESARROLLO DE TAREAS:

- ✓ **Definir y generar tabla business, contact y sus relaciones.**

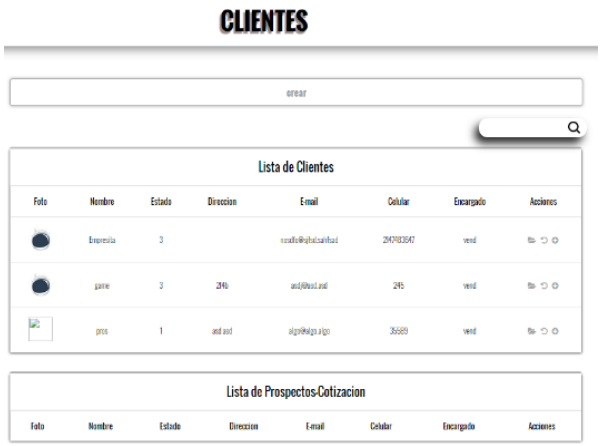


Grafica 23 Tabla business, contact y sus relaciones

BUSINESS			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
	Address	varchar(255)	Dirección de residencia actual
	City	varchar(255)	Ciudad de residencia actual
	Country	varchar(255)	País de residencia actual
	Email	varchar(255)	Correo electronico
	Expedition_date	date	Fecha de registro
	Ext	int(11)	Extención del número telefonico
	Fax	int(11)	Numero de fax
	Maps	varchar(255)	Localizacion en el mapa
	Mobile_phone	int(11)	Numero de telefono movil
	Name	varchar(255)	Nombre
	Nit	int(11)	Numero de nit
	Page_web	varchar(255)	URL de pagina web
	Payment_date	date	Fecha de pago
	Phone	int(11)	Numero telefonico
	Photo	varchar(255)	Ruta archivo de imagen -Logo Empresa-
	Second_email	varchar(255)	Segundo Correo eletronico
	Skype	varchar(255)	Nombre de usuario de Skype
	Source	varchar(255)	Fuente del contacto
	State	int(11)	Determina en qué estado se encuentra
	Type	int(11)	Determina si es cliente o prospecto
FK	Manager	int(11)	Id del usuario encargado

CONTACT			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Numero de identificacion
	Charge	varchar(255)	Nombre del cargo
	Email	varchar(255)	Correo electronico
	Last_name	varchar(255)	Apellido
	Mobile_phone	int(11)	Numero de telefono movil
	Name	varchar(255)	Nombre
	Phone	int(11)	Numero telefonico
FK	Business_id	int(11)	Precio del producto vendido a la empresa

- ✓ **Elaboración de la plantilla de control de clientes y prospectos.**



Grafica 24 Plantilla control clientes y prospectos

- ✓ **Generar código de organización de información dentro de la plantilla.**

```
<div class="wrapperContent">
  <section class="tableContent">
    <h2>Clientes</h2>
    <table class=" ">
      <thead>
        <tr>
            <th>Foto</th>
            <th>Nombre</th>
            <th>Estado</th>
            <th>Direccion</th>
            <th>E-mail</th>
            <th>Celular</th>
            <th>Encargado</th>
            <th>Acciones</th>
        </tr>
      </thead>

      <tbody>
      @foreach($business as $businessClient)
        @if($businessClient->type==1)
          <tr>
            <td>{{ HTML::image('business/'.$businessClient->photo,'',array('style'=>'width: 100px;')) }}</td>
            <td>{{$businessClient->name}}</td>
            <td>{{$businessClient->state}}</td>
```

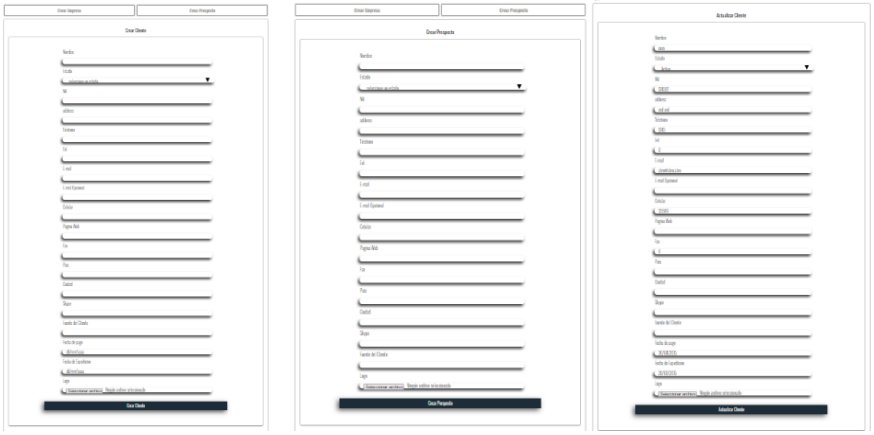
```
              <td>{{$businessClient->address}}</td>
              <td>{{$businessClient->email}}</td>
              <td>{{$businessClient->mobile_phone}}</td>
              @foreach($users as $user)
                @if($user->id==$businessClient->manager)
                  <td>{{$user->user_name}}</td>
                @endif
          @endforeach

              <td><a class="icon-folder-open" href="{{route('seeBusiness',$businessClient->id)}}"></a> <a href="{{route
('updateBusiness',$businessClient->id)}}" class="icon-ccw "></a><a class="icon-plus-circled" href="{{route('createProducts',
$businessClient->id)}}"></a></td>
            </tr>
          @endif
            @endforeach

       </tbody>
    </table>
  </section>
</div>

<div class="wrapperContent">
  <section class="tableContent">
    <h2>Prospectos</h2>
      <table class=" ">
        <thead>
          <tr>
              <th>Foto</th>
              <th>Nombre</th>
              <th>Estado</th>
              <th>Direccion</th>
              <th>E-mail</th>
              <th>Celular</th>
              <th>Encargado</th>
              <th>Acciones</th>
           </tr>
         </thead>
         <tbody>
         @foreach($business as $businessClient)
           @if($businessClient->type==2)
             <tr>
               <td>{{ HTML::image('business/'.$businessClient->photo,'',array('style'=>'width: 100px;')) }}</td>
               <td>{{$businessClient->name}}</td>
               <td>{{$businessClient->state}}</td>
               <td>{{$businessClient->address}}</td>
               <td>{{$businessClient->email}}</td>
               <td>{{$businessClient->mobile_phone}}</td>
               @foreach($users as $user)
                 @if($user->id==$businessClient->manager)
                   <td>{{$user->user_name}}</td>
                 @endif
              @endforeach
                <td><a       class="icon-folder-open"       href="{{route('seeBusiness',$businessClient->id)}}"></a>       <a
href="{{route('updateBusiness',$businessClient->id)}}" class="icon-ccw "></a></td>
             </tr>
           @endif
         @endforeach
       </tbody>
    </table>
  </section>
</div>
```

✓ **Elaboración de formularios de registro y modificación.**

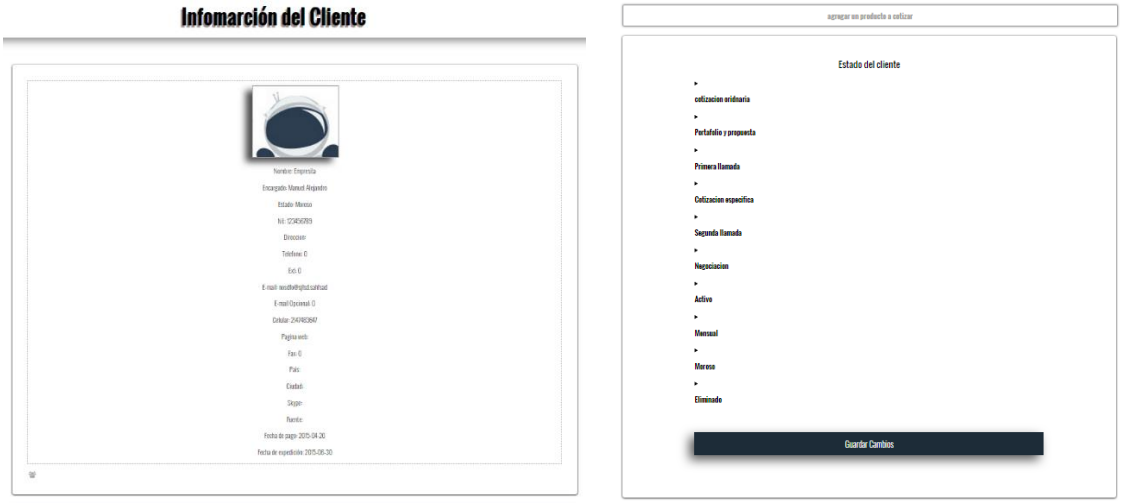


Grafica 25 Formularios control cliente y prospecto

✓ **Generar el código de mensajes, captura de datos y re direccionamiento.**

```
public function saveBusiness()
{
        $businessManager=new BusinessManager(new Business(),Input::all());
        $businessValidator=$businessManager->isValid();
        if($businessValidator)
        {
      return Redirect::route('business')->with('type', Input::get('type'))->withErrors($businessValidator)->withInput();
        }
        $businessName=$businessManager->saveBusiness();
        if($businessName){
          new LogRepo(
            [
                    'responsible'=> Auth::user()->user_name,
                    'responsible_id'=> Auth::user()->id,
                    'action' => 'ha creado un '.$businessName,
                    'affected_entity'=> Input::get('name'),
            ]
          );
          return Redirect::route('business')->with('message','el '.$businessName.' fue creado exitosamente');
        }
        return Redirect::route('business')->with('message_error','el '.$businessName.' no pudo ser creado');
}
```

✓ Elaboración de la plantilla de información.



Grafica 26 Plantilla de información

✓ Generar código de organización de información dentro de la plantilla.

```
<section>
          <figure>
             {{ HTML::image('business/'.$business->photo,'',array('style'=>'width: 100px;')) }}
          </figure>
          <section>
             <p>Nombre: {{$business->name}}</p>
             <p>Encargado: {{$manager->name }}</p>
             <p>Estado: {{$state}}</p>
             <p>Nit: {{$business->nit}}</p>
             <p>Direccion: {{$business->address}}</p>
             <p>Telefono: {{$business->phone}}</p>
             <p>Ext: {{$business->ext}}</p>
             <p>E-mail: {{$business->email}}</p>
             <p>E-mail Opcional: {{$business->ext}}</p>
             <p>Celular: {{$business->mobile_phone}}</p>
             <p>Pagina web: {{$business->page_web}}</p>
             <p>Fax: {{$business->fax}}</p>
             <p>Pais: {{$business->country}}</p>
             <p>Ciudad: {{$business->city}}</p>
             <p>Skype: {{$business->skype}}</p>
             <p>Fuente: {{$business->source}}</p>
             @if($business->type==1)
             <p>Fecha de pago: {{$business->payment_date}}</p>
             <p>Fecha de expedición: {{$business->expedition_date}}</p>
             @endif
          </section>
</section>
@if($business->maps!='')
<div id="map-canvas"></div>
@endif
<a class=" icon-users buttonMaps" ></a>
<section class="business-states">
    <p class="get-type hidden">{{$business->type}}</p>
    <p class="get-state hidden">{{$business->state}}</p>
    <div class="formContent">
    <h2>Estado del cliente</h2>
          {{Form::open(array('name'=>'form-see-business','url'=>  'admin/verEmpresa/'.$business->id,  'method'  =>
'POST')) }}
    <details class="state-input hidden">
        <summary><h3>cotizacion oridnaria</h3></summary>
```

```
        <p class="state-p0">Comentarios: {{$business->record["state_one"]}}</p>
        {{Form::textarea('state_one',$business->record["state_one"],['class'=>'state0','id'=> 'hidden']) }}
            </details>
            <details class="state-input hidden">
               <summary><h3>Portafolio y propuesta</h3></summary>
               <p class="state-p1">Comentarios: {{$business->record["state_two"]}}</p>
               {{ Form::textarea('state_two',$business->record["state_two"], ['class' => 'state1 ','id'=>'hidden']) }}
            </details>
            <details class="state-input hidden">
               <summary><h3>Primera llamada</h3></summary>
               <p class="state-p2">Comentarios: {{$business->record["state_three"]}}</p>
               {{ Form::textarea('state_three',$business->record["state_three"], ['class' => 'state2 ','id'=>'hidden']) }}
            </details>
            <details class="state-input hidden">
               <summary><h3>Cotizacion especifica</h3></summary>
               <p class="state-p3">Comentarios: {{$business->record["state_four"]}}</p>
               {{ Form::textarea('state_four',$business->record["state_four"], ['class' => 'state3 ','id'=>'hidden']) }}
            </details>
            <details class="state-input hidden">
               <summary><h3>Segunda llamada</h3></summary>
               <p class="state-p4">Comentarios: {{$business->record["state_five"]}}</p>
               {{ Form::textarea('state_five',$business->record["state_five"], ['class' => 'state4 ','id'=>'hidden']) }}
            </details>
            <details class="state-input hidden">
               <summary><h3>Negociacion</h3></summary>
               <p class="state-p5">Comentarios: {{$business->record["state_six"]}}</p>
               {{ Form::textarea('state_six',$business->record["state_six"], ['class' => ' state-change','id'=>'hidden']) }}
            </details>
            <details class="state-input hidden">
               <summary><h3>Activo</h3></summary>
               <p class="state-p6">Comentarios: {{$business->record["state_seven"]}}</p>
               {{ Form::textarea('state_seven',$business->record["state_seven"], ['class' => 'state6 ','id'=>'hidden']) }}
            </details>
            <details class="state-input hidden">
               <summary><h3>Moroso</h3></summary>
               <p class="state-p8">Comentarios: {{$business->record["state_nine"]}}</p>
               {{ Form::textarea('state_nine',$business->record["state_nine"], ['class' => 'state8 ','id'=>'hidden']) }}
            </details>
            <details class="state-input hidden">
               <summary><h3>Eliminado</h3></summary>
               <p class="state-p9">Comentarios: {{$business->record["state_ten"]}}</p>
               {{ Form::textarea('state_ten',$business->record["state_ten"] ,['class' => 'state9 ','id'=>'hidden']) }}
            </details>
            {{ Form::submit('Guardar Cambios') }}
            {{ Form::close() }}
          </div>
      </section>
```

- ✓ **Generar código de inserción de datos de cliente en formulario para actualización.**

```
public function showUpdateBusiness($id)
{
      $updateBusiness=Business::find($id);
      $permiso =new Proceso();
      $total=$permiso->filtrarPermisos();
      $business=Business::all();
      $users=User::all();
      $userRepo=new UserRepo();
      $managers=$userRepo->getManager(Auth::user()->id,Auth::user()->role_id,Auth::user()->user_name);
      $state=$userRepo->getState();
      $states=$userRepo->getStates();
      return
View::make('front.business.updateBusiness',compact('business','total','managers','state','states','users','updateBusiness'));
}
```

✓ **Elaboración de la plantilla de asociación de productos.**



Grafica 27 Asociación de productos

✓ **Generar el código de mensajes, captura de datos y re direccionamiento.**

```
public function addProducts($id)
{
      $productManager=new CreateProductManager(new BusinessProduct(),Input::all());
      $productValidator=$productManager->isValid();
      if($productValidator)
      {
         return Redirect::route('createProducts',$id)->withErrors($productValidator)->withInput();
      }
      $productManager->saveProduct($id);
      new LogRepo(
         [
            'responsible'=> Auth::user()->user_name,
            'responsible_id'=> Auth::user()->id,
            'action' => 'ha agregado un producto a una empresa ',
            'affected_entity'=> '',
         ]
      );
      return Redirect::route('createProducts',$id)->with('message','el producto fue creado correctamente');
}
```

PRUEBAS DE ACEPTACIÓN:

✓ **Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.**

Todas las advertencias y mensajes son recibidos correctamente

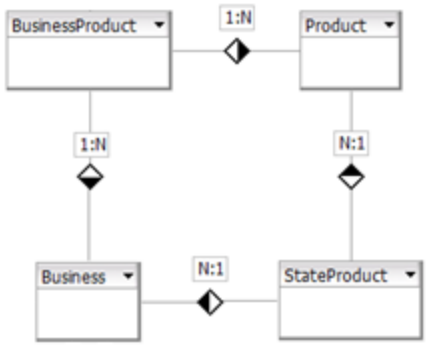
- ✓ **Asociación de cliente con el producto satisfactoria.**

La asociación es creada correctamente.

- **CONTROLAR PRODUCTOS**

DESARROLLO DE TAREAS:

- ✓ **Definir y generar tabla product y sus relaciones.**



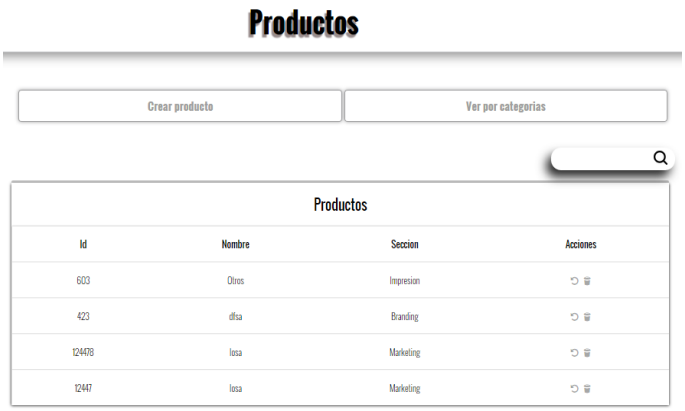
Grafica 28 Tabla product y sus relaciones

PRODUCT			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Numero de identificacion
	Dependence	varchar(255)	Nombre de la dependencia
	Name	varchar(255)	Nombre

BUSINESSPRODUCT			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Numero de identificacion
	Value	int(11)	Precio del producto vendido a la empresa
FK	Business_id	int(11)	Identificacion de la empresa
FK	Product_id	int(11)	Identificacion del producto

STATEPRODUCT			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
	Quantity	int(11)	Cantidad de productos
	Value	int(11)	Valor del producto
FK	Business_id	int(11)	Identificación de la empresa
FK	Product_id	int(11)	Identificación del producto

✓ **Elaboración de la plantilla de control de productos.**

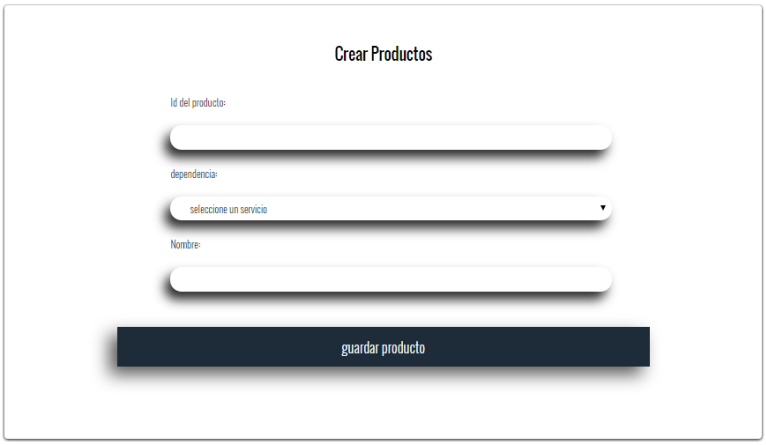


Grafica 29 Plantilla control de productos

✓ **Generar código de organización de información dentro de la plantilla.**

```
<section class="tableContent">
        <h2>Productos</h2>
        <table class=" ">
           <thead>
           <tr>
              <th>Id</th>
              <th>Nombre</th>
              <th>Seccion</th>
              <th>Acciones</th>
           </tr>
           </thead>
           <tbody>
           @foreach($products as $product)
                 <tr>
                    <td>{{$product->id}}</td>
                    <td>{{$product->name}}</td>
                    @if($product->dependency==1)
                       <td>Branding</td>
                    @endif
                    @if($product->dependency==2)
                       <td>Web y App</td>
                    @endif
                    @if($product->dependency==3)
                       <td>Marketing</td>
                    @endif
                    @if($product->dependency==4)
                       <td>Produccion Audiovisual</td>
                    @endif
                    @if($product->dependency==5)
                       <td>Estrategia</td>
                    @endif
                    @if($product->dependency==6)
                       <td>Impresion</td>
                    @endif
                    <td><a class="icon-ccw " href="{{route('updateProduct',$product->id)}}"></a><a  class="icon-trash "
href="{{route('deleteProduct',$product->id)}}"></a></td>
                 </tr>
           @endforeach
           </tbody>
        </table>
</section>
```

- ✓ **Elaboración de formulario de agregar producto.**



Grafica 30 Formulario agregar producto

- ✓ **Generar el código de mensajes, captura de datos y re direccionamiento.**

```
public function saveProduct()
{
      $productManager=new ProductManager(new Product(),Input::all());
      $productValidator=$productManager->isValid();
      if($productValidator)
      {
         return Redirect::route('product')->withErrors($productValidator)->withInput();
      }
      $productManager->saveProduct();
      new LogRepo(
         [
            'responsible'=> Auth::user()->user_name,
            'responsible_id'=> Auth::user()->id,
            'action' => 'ha creado un producto ',
            'affected_entity'=> Input::get('name'),
         ]
      );
      return Redirect::route('product')->with('message','el producto se guardó correctamente');
}
```

PRUEBAS DE ACEPTACIÓN:

- ✓ **Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.**

Todas las advertencias y mensajes son recibidos correctamente

- **SUPERVISAR USUARIOS**

DESARROLLO DE TAREAS:

✓ Elaboración de la plantilla de administración.



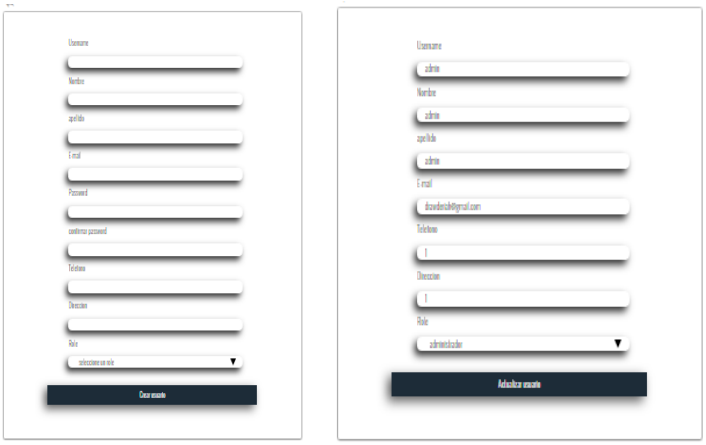
Grafica 31 Plantilla administración

✓ Generar código de organización de información dentro de la plantilla.

```
<table class="tableContent">
        <thead>
        <tr>
           <th>Foto</th>
           <th>Nombre</th>
           <th>Apellido</th>
           <th>Correo</th>
           <th>Direccion</th>
           <th>Telefono</th>
           <th>Role</th>
           <th>Encargado</th>
           <th>Acciones</th>
        </tr>
        </thead><tbody>
        @foreach ($users as $user)
           <tr>
              <td>{{ HTML::image('user/'.$user->photo,'',array('id'=>'')) }}</td>
              <td>{{$user->name}}</td>
              <td>{{$user->last_name}}</td>
              <td>{{$user->email}}</td>
              <td>{{$user->address}}</td>
              <td>{{$user->phone}}</td>
              @if($user->role_id==0)
                 <td>sin cargo</td>
              @endif
              @if($user->role_id==1)
                 <td>Super Administrador</td>
              @endif
              @if($user->role_id==2)
                 <td>Administrador</td>
              @endif
```

```
            @if($user->role_id==3)
                <td>Vendedor</td>
            @endif
            <p class="i hidden">{{$i=0}}</p>
            @foreach($users as $user1)
                @if($user1->id==$user->manager)
                    <p class="i hidden">{{$i=1}}</p>
                    <td>{{$user1->user_name}}</td>
                @endif
            @endforeach
            @if($i==0)
                <td>sin encargado</td>
            @endif
            @if(isset($total[7]))
                @if($total[7]==1)
                <td><a href="{{route('showUser',$user->id)}}" class="icon-folder-open "></a>
                @endif
            @endif
            @if(isset($total[6]))
                @if($total[6]==1)
                    <a href="{{route('updateUser',$user->id)}}" class="icon-ccw "></a>
                @endif
            @endif
            @if(isset($total[5]))
                @if($total[5]==1)
                  <a href="{{route('deleteUser',$user->id)}}" class="icon-trash "></a></td>
                @endif
            @endif
          </tr>
        @endforeach
        </tbody>
</table>
```

✓ Elaboración de formularios de registro y modificación.

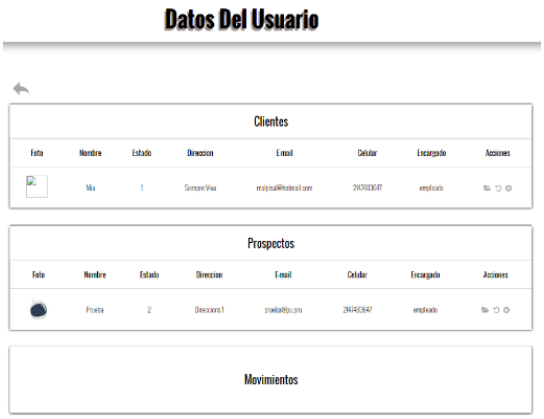


Grafica 32 Formularios supervisión de usuarios

✓ **Generar el código de mensajes, captura de datos y re direccionamiento.**

```
public function createUser()
{
      $permiso =new Proceso();
      $total=$permiso->filtrarPermisos();
      if(Auth::user()->role_id==1)
      {
         $charges= [ 'seleccione un role'=> "seleccione un role"]
            +[ 3 => "vendedor"]
            +[ 2 => "administrador"]
            +[ 1 => "super administrador"];
         $managers = [ "0" => "selecione un encargado"]
            +User::whereRaw('role_id=2')->lists('user_name','id');
      }
      if(Auth::user()->role_id==2)
      {
         $charges= [ 'seleccione un role'=> "seleccione un role"]
            +[ 3 => "vendedor"];
         $managers = [ "0" => "selecione un encargado"]
            +[''.Auth::user()->id.''=>''.Auth::user()->user_name.''];
      }
      if(Auth::user()->role_id==3)
      {
         $charges= [ 'seleccione un role'=> "seleccione un role"];
         $managers = [ "0" => "selecione un encargado"];
      }
      return View::make('front.accounts.create',compact('managers','charges','total'));
}
```

✓ **Elaboración de la plantilla de información.**



Grafica 33 Plantilla información usuario

✓ **Generar código de organización de información dentro de la plantilla.**

```
<a class="icon-reply back" href="{{route('administrator')}}"></a>
        @if($user->role_id==3)
        <div class="wrapperContent">
        <div class="tableContent ">
           <h2>Clientes</h2>
           <table class=" show-business">
              <thead>
              <tr>
```

```
                <th>Foto</th>
                <th>Nombre</th>
                <th>Estado</th>
                <th>Direccion</th>
                <th>E-mail</th>
                <th>Celular</th>
                <th>Encargado</th>
                <th>Acciones</th>
            </tr>
            </thead>
            <tbody>
            @foreach($business as $businessClient)
                @if($businessClient->type==1)
                    <tr>
                        <td>{{ HTML::image('business/'.$businessClient->photo,'',array('style'=>'width: 100px;')) }}</td>
                        <td>{{$businessClient->name}}</td>
                        <td>{{$businessClient->state}}</td>
                        <td>{{$businessClient->address}}</td>
                        <td>{{$businessClient->email}}</td>
                        <td>{{$businessClient->mobile_phone}}</td>
                        <td>{{$user->user_name}}</td>
                        <td><a  class="icon-folder-open"  href="{{route('seeBusiness',$businessClient->id)}}"></a>  <a
href="{{route('updateBusiness',$businessClient->id)}}"  class="icon-ccw  "></a><a  class="icon-plus-circled"
href="{{route('createProducts',$businessClient->id)}}"></a></td>
                    </tr>
                @endif
            @endforeach
            </tbody>
        </table>
        </div>
        </div>
    <div class="wrapperContent">
        <div class="tableContent">
            <h2>Prospectos</h2>
            <table class="show-business ">
            <thead>
            <tr>
                <th>Foto</th>
                <th>Nombre</th>
                <th>Estado</th>
                <th>Direccion</th>
                <th>E-mail</th>
                <th>Celular</th>
                <th>Encargado</th>
                <th>Acciones</th>
            </tr>
            </thead>
            <tbody>
            @foreach($business as $businessClient)
                @if($businessClient->type==2)
                    <tr>
                        <td>{{ HTML::image('business/'.$businessClient->photo,'',array('style'=>'width: 100px;')) }}</td>
                        <td>{{$businessClient->name}}</td>
                        <td>{{$businessClient->state}}</td>
                        <td>{{$businessClient->address}}</td>
                        <td>{{$businessClient->email}}</td>
                        <td>{{$businessClient->mobile_phone}}</td>
                        <td>{{$user->user_name}}</td>
                        <td><a  class="icon-folder-open"  href="{{route('seeBusiness',$businessClient->id)}}"></a>  <a
href="{{route('updateBusiness',$businessClient->id)}}"  class="icon-ccw  "></a><a  class="icon-plus-circled"
href="{{route('createProducts',$businessClient->id)}}"></a></td>
                    </tr>
                @endif
            @endforeach
            </tbody>
        </table>
    </div>
</div>
@endif
    <div>
        @if(count($logs)>0)
            <h2 class="showMove">Movimientos</h2>
            <section class="profileLogs hidden">
                @foreach($logs as $log)
                    <p>
                        {{$log->responsible}}
```

```
                        {{$log->action}}
                        @if($log->affected_entity)
                           de {{$log->affected_entity}}
                        @endif
                        el {{$log->created_at}}
                     </p>
                  @endforeach
               </section>  @endif
         </div>
```

- ✓ **Generar código de inserción de datos de cliente en formulario para actualización.**

```
   <a class="icon-reply back" href="{{route('administrator')}}"></a>
   <div class="formContent">
      {{ Form::open(array('name'=>'form-create-user','to' => 'admin/actualizarUsuario/'.$user->id, 'method' => 'POST'))
}}
      {{ Form::text('id',$user->id,['id'=>'hidden']) }}
      <div>
         {{ Form::label('user_name', 'Username') }}
         {{ Form::text('user_name',$user->user_name) }}
         {{$errors->first('user_name')}}
      </div>
      <div>
         {{ Form::label('name', 'Nombre') }}
         {{ Form::text('name',$user->name) }}
         {{$errors->first('name')}}
      </div>
      <div>
         {{ Form::label('last_name', 'apellido') }}
         {{ Form::text('last_name',$user->last_name) }}
         {{$errors->first('last_name')}}
      </div>
      <div>
         {{ Form::label('email', 'E-mail') }}
         {{ Form::text('email',$user->email) }}
         {{$errors->first('email')}}
      </div>
      <div>
         {{ Form::label( 'phone','Telefono') }}
         {{ Form::text('phone',$user->phone) }}
         {{$errors->first('phone')}}
      </div>
      <div>
         {{ Form::label('address', 'Direccion') }}
         {{ Form::text('address',$user->address) }}
         {{$errors->first('address')}}
      </div>
      <div>
         {{ Form::label('role_id', 'Role') }}
         {{ Form::select('role_id', $charges, $user->role_id, array('id' => 'role_id')) }}
         {{$errors->first('role_id')}}
      </div>
      <div class="create-manager">
         {{Form::label('manager','Encargado','',array('class'=>'manager'))}}
         {{ Form::select('manager', $managers, $user->manager, array('id' => 'manager')) }}
         {{ Session::get('manager','',array('class'=>'manager')) }}
         {{$errors->first('manager')}}
      </div>
      {{ Form::submit('Actualizar usuario') }}
      {{ Form::close() }}
   </div>
```

PRUEBAS DE ACEPTACIÓN:

- ✓ **Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.**

Todas las advertencias y mensajes son recibidos correctamente

- ✓ **Asociación bien asignada.**

La asociación es creada correctamente.

- **BUSCAR EN EL SISTEMA**

DESARROLLO DE TAREAS:

- ✓ **Elaboración de la barra de búsqueda.**



Grafica 34 Barra de Búsqueda

Código CSS

```
.formSearch{
   width: 100%;
}

.formSearch button{
   width: 26px;
   float: right;
   margin: -27px 10px;
   position: relative;
   height: 24px;
   background: white;
   border: none;
}

.buttonSearch{
   margin-left: 80%;
   width: 200px;
}

.buttonSearch img{
   width: 20px;
   float: right;
   margin: -11px -4px;
   position: relative;
}
```

- ✓ **Generar código de búsqueda.**

```
public function search()
   {
      $permiso =new Proceso();
      $total=$permiso->filtrarPermisos();

      //nombreTabla = ConsultaSQL; Esto se hace para todas las tablas que
        participen dentro de la búsqueda

      if(Input::get('search')=="")
      {
            $nombreTabla= nombreTabla::all();
      }
      return View::make(NombreVista,compact(Datos));
   }
```

- ✓ **Generar código de obtención y organización de datos.**

Como se explicó anteriormente así como la plantilla es la misma de tablas el código de organización de datos también es el mismo.

PRUEBAS DE ACEPTACIÓN:

- ✓ **Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.**

Todas las advertencias y mensajes son recibidos correctamente

- ✓ **Datos obtenidos correctamente.**



Grafica 35 Resultados búsqueda

Los resultados son correctos.

- **COMUNICACIÓN EN EL SISTEMA**

DESARROLLO DE TAREAS:

✓ Elaboración de formulario de envío de correos.



Grafica 36 Formulario envió de correo

✓ Generar el código de mensajes, captura de datos y re direccionamiento.

```
public function sendContact()
{
      $data=Input::all()
         +['name'=>Auth::user()->name]
         +['user_name'=>Auth::user()->user_name]
         +['email'=>Auth::user()->email]
         +['about'=>Input::get('about')]
         +['text'=>Input::get('message')]
         +['link'=>Auth::user()->id];
      if(Auth::user()->role_id==3)
      {
         $manager=User::find(Auth::user()->manager);
         $email=$manager->email;
         Mail::send('emails.contactAs', $data, function ($message) use ($email) {
            $message->to($email, 'cerveruss CRM')->subject('correo de '.Auth::user()->user_name.'');
         });
         return Redirect::route('contactAs')->with('message','El correo se envio correctamente al Administrador a su
cargo');
      }
      if(Auth::user()->role_id==2)
      {
         $users=User::where('role_id','=',1)->get();
         foreach($users as $user)
         {
            $email=$user->email;
            Mail::send('emails.contactAs', $data, function ($message) use ($email) {
               $message->to($email, 'cerveruss CRM')->subject('correo de '.Auth::user()->user_name.'');
            });
            return Redirect::route('contactAs')->with('message','El correo se envio correctamente a los super
administradores');
```

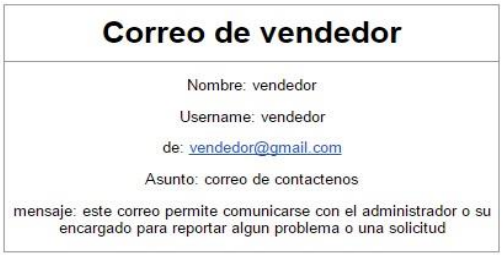
```
        }
    }
    return Redirect::route('contactAs')->with('message_error','el Super Administrador no puede enviar correos');
}
```

PRUEBAS DE ACEPTACIÓN:

- ✓ **Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.**

Todas las advertencias y mensajes son recibidos correctamente

- ✓ **Correo enviado y recibido satisfactoriamente.**



Grafica 37 Correo satisfactorio

- **SEGUIMIENTO DE CLIENTE Y PROSPECTO**

DESARROLLO DE TAREAS:

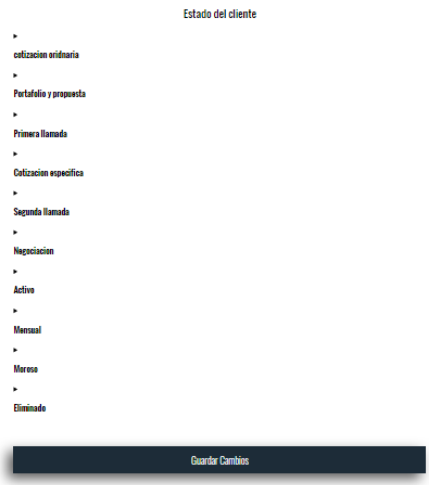
- ✓ **Definir y generar tabla record y sus relaciones.**



Grafica 38 Tabla record y sus relaciones

RECORD			
LLAVE	**NOMBRE**	**TIPO**	**DESCRIPCION**
PK	Id	int(11)	Número de identificación
	State_one	varchar(255)	Estado 1 de la empresa
	State_two	varchar(255)	Estado 2 de la empresa
	State_three	varchar(255)	Estado 3 de la empresa
	State_four	varchar(255)	Estado 4 de la empresa
	State_five	varchar(255)	Estado 5 de la empresa
	State_six	varchar(255)	Estado 6 de la empresa
	State_seven	varchar(255)	Estado 7 de la empresa
	State_eight	varchar(255)	Estado 8 de la empresa
	State_nine	varchar(255)	Estado 9 de la empresa
	State_ten	varchar(255)	Estado 10 de la empresa
FK	Business_id	int(11)	Identificación de la empresa

- ✓ **Elaboración de formulario de envío de correos.**



Grafica 39 Seguimiento cliente

- ✓ **Generar el código de mensajes, captura de datos y re direccionamiento.**

```
public function saveStates($id)
{
      $business=Business::find($id);
      $business->record["state_one"]=Input::get("state_one");
      $business->record["state_two"]=Input::get("state_two");
      $business->record["state_three"]=Input::get("state_three");
      $business->record["state_four"]=Input::get("state_four");
      $business->record["state_five"]=Input::get("state_five");
      $business->record["state_six"]=Input::get("state_six");
      $business->record["state_seven"]=Input::get("state_seven");
      $business->record["state_eight"]=Input::get("state_eight");
      $business->record["state_nine"]=Input::get("state_nine");
      $business->record["state_ten"]=Input::get("state_ten");
      if($business->record->save())
      {
         new LogRepo(
            [
```

```
                'responsible'=> Auth::user()->user_name,
                'responsible_id'=> Auth::user()->id,
                'action' => 'ha modificado los estados de la empresa ',
                'affected_entity'=> $business->name,
            ]
        );
        $businessRepo=new BusinessRepo();
        $businessRepo->saveState($business->type,$business->state,Input::all(),$id);
        return Redirect::route('seeBusiness',$id)->with("message",'se guardo los estados correctamente');
    }else{
        return Redirect::route('seeBusiness',$id)->with("message_error",'no se guardo los estados correctamente');
    }
}
```

PRUEBAS DE ACEPTACIÓN:

- ✓ **Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.**

Todas las advertencias y mensajes son recibidos correctamente

- ✓ **El prospecto queda registrado como cliente.**

Se ha verificado y el prospecto queda registrado como cliente.

- **MARKETING**

DESARROLLO DE TAREAS:

- ✓ **Creación formulario de marketing.**



Grafica 40 Marketing

- ✓ **Generar el código de mensajes, captura de datos y re direccionamiento.**

```
public function sendPromotion()
{
      $business=Business::all();
      $data=[];
      foreach($business as $client)
      {
         $email=$client->email;
         Mail::send('emails.auth.reminder', $data, function ($message) use ($email) {
            $message->to($email, 'cerveruss CRM')->subject('correo de Mi-Martinez.com');
         });
      }
      return Redirect::route('promotion')->with('message','los correos fueron enviados exitosamente');
}
```

PRUEBAS DE ACEPTACIÓN:

- ✓ **Recibir advertencia o mensajes de error al ingresar en los formularios datos no permitidos o dejar en blanco campos obligatorios.**

Todas las advertencias y mensajes son recibidos correctamente

- ✓ **Correo enviado y recibido satisfactoriamente.**



Grafica 41 Correo satisfactorio 2

- **CONSULTAR REPORTES**

DESARROLLO DE TAREAS:

✓ Elaboración del diagrama circular con api de google para clientes.

El diagrama circular se obtiene gracias a la librearía de APIs de google y se les entrega los datos como se muestra en el siguiente código:

```
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
   <script type="text/javascript">
      // Load the Visualization API and the piechart package.
      google.load('visualization', '1.0', {'packages':['corechart']});
      // Set a callback to run when the Google Visualization API is loaded.
      google.setOnLoadCallback(drawChart);
      // Callback that creates and populates a data table,
      // instantiates the pie chart, passes in the data and
      // draws it.

      function drawChart() {

         // Create the data table.
         var data = new google.visualization.DataTable();
         var business=$('.countBusiness').val();
         var client=$('.countClient').val();
         var recurrent=$('.countRecurrent').val();
         data.addColumn('string', 'Topping');
         data.addColumn('number', 'Slices');
         data.addRows([
            ['('+business+') Clientes', parseInt(business)],
            ['('+client+') Prospectos', parseInt(client)],
            ['('+recurrent+') Recurrentes', parseInt(recurrent)]
         ]);

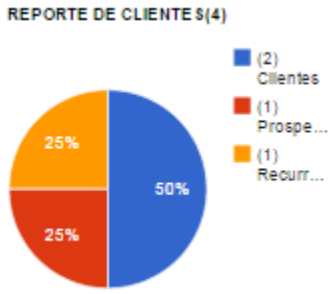
         // Set chart options
         var total=parseInt(business)+ parseInt(client);
         var options = {'title':'REPORTE DE CLIENTES('+total+')',
            'width':500,
            'height':400};

         // Instantiate and draw our chart, passing in some options.
         var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
         chart.draw(data, options);
      }
   </script>
```

Así asignándole a una estructura HTML el diagrama circular.

```
   <section class="reports">
      <div id="chart_div">
         <input class="countBusiness hidden" value="{{count($business)}}">
         <input class="countClient hidden" value="{{count($client)}}">
         <input class="countRecurrent hidden" value="{{$recurrent}}">
      </div>
   </section>
```

Dando como resultado lo ilustrado en la Grafica 42:



Grafica 42 Diagrama Circular clientes

- ✓ **Creación tabla de clientes asignados.**

Vendedor	Clientes	Prospecto	Productos Vendidos
vendedor vendedor	2	2	2

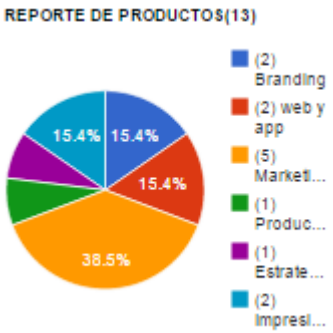
Grafica 43 Tabla clientes asignados

- ✓ **Generar el código de captura de datos.**

```
public function show()
   {
      $permiso =new Proceso();
      $total=$permiso->filtrarPermisos();
      $business=Business::where('type','=',1)->get();
      $client=Business::where('type','=',2)->get();
      $recurrent=0;
      foreach($client as $dates)
      {
         $more=$this->date($dates->created_at);
         if($more)
         {
            $recurrent++;
         }
      }
      $users=User::where('role_id','=',3)->get();
      foreach($users as $user)
      {
         $user->role_id=count(Business::where('manager','=',$user->id)->get());
      }
      return View::make('front.reports.chart',compact('total','business','client','recurrent','users'));
   }
```

- ✓ **Elaboración del diagrama circular con api de google para productos.**

Como se declaró anteriormente solo se tienen que entregar los datos al api para generar el diagrama.



Grafica 44 Diagrama circular productos

- ✓ **Creación tablas de reportes.**

Dependencia	Nombre del producto	Cliente	Valor del producto
BRANDING	Diseño de Imagen Corporativa	prospectocliente	$ 12313
BRANDING	Diseño de Imagen Corporativa	prospectocliente	$ 432
WEB	Sitio Basico	Prueba	$ 4324
WEB	Especial	prospectocliente	$ 321423
MARKETING	Mail Marketing	prospectocliente	$ 432
MARKETING	Social Media	Prueba	$ 4324

Grafica 45 Tabla productos vendidos

- ✓ **Generar el código de captura de datos.**

```
public function show()
   {
      $permiso =new Proceso();
      $total=$permiso->filtrarPermisos();
      $business=Business::where('type','=',1)->get();
      $prodcut=Business::where('type','=',2)->get();
      $recurrent=0;
      foreach($product as $dates)
      {
         $more=$this->date($dates->created_at);
         if($more)
         {
            $recurrent++;
         }
      }
      $users=User::where('role_id','=',3)->get();
      foreach($users as $user)
      {
         $user->role_id=count(Business::where('manager','=',$user->id)->get());
      }
      return View::make('front.reports.chart',compact('total','business','product','recurrent','users'));
   }
```

- ✓ **Tablas de consulta de estado del cliente.**

Clientes	Prospecto	Estado
cliente	--	Moroso
--	prospecto	Primera llamada
client	--	Moroso
--	prospectos	Cotización oridinaria
maps	--	Moroso
--	mapsp	Cotización oridinaria

cotizacion oridnaria	Portafolio y propuesta	Primera llamada	Cotizacion especifica	Segunda llamada	Negociación
---	---	prospecto	---	---	---
prospectos	---	---	---	---	---
mapsp	---	---	---	---	---

Activo	Mesual	Moroso	Eliminado
---	---	cliente	---
---	---	client	---
---	---	maps	---

PRUEBAS DE ACEPTACION:

- ✓ **Reportes son precisos y correctos.**

Como se muestra en las Gráficas 42, 43, 44 y 45 los datos son entregados correctamente y se comprobó su precisión.

4. CONCLUSIONES

- En el ámbito de desarrollo WEB actual es esencial el desarrollar aplicaciones con la ayuda de frameworks como “Node.js”, “Ruby on Rails”, “Django” o en este caso de “Laravel” que otorgan una mayor rapidez a la construcción de módulos, adaptabilidad y una estructura que permite sencillez a la hora de codificar el software, además de soportar múltiples librerías y permitir la conexión a múltiples motores de base de datos.

- Al momento de desarrollar es fundamental comprender lo que se quiere realizar antes de comenzar el desarrollo, tener una organización clara que todo el equipo entienda y pueda cumplir con sus responsabilidades sin la necesidad de supervisión constante para ello se emplean metodología de desarrollo de software que permiten una estructura, planificación y control organizado que logra cumplir con los objetivos del desarrollo. En este proyecto fue utilizada la metodología ágil SCRUM, que permitió una mayor rapidez y fluidez a la hora de realizar el software, pero para utilizarla es fundamental el entender las bases de las metodologías.

- Las empresas de desarrollo actuales necesitan velocidad de desarrollo de esta forma poder cumplir con las exigencias del cliente, además de dejar abierto el software a nuevas actualización por lo que es necesario utilizar técnicas de adaptabilidad. Comprendiendo que es necesario una constante capacitación y saber que podemos y no lograr, ya que al estar en constate contacto con el cliente pueden surgir cambios a los cuales debemos negarnos o rápidamente ser capaces de desarrollar e integrar al software.

- Al utilizar las tecnologías actuales de desarrollo, se observa que constantemente (3 a 6 meses) se efectúan actualizaciones, por lo que no hacer una buena investigación y estar constantemente enterado, es un error que podría causar problemas en el desarrollo del proyecto.

5. RECOMENDACIONES

- Adaptar el aplicativo para que funcione en Laravel 5, ya que contiene una estructuración que facilita la búsqueda y rapidez del aplicativo y deja utilizar de forma muy sencilla los namespaces que sirven para crear tu propio modelo de negocios según los requerimientos y necesidades del programa entre otras actualizaciones que contiene Laravel 5.

- Creación de un chat interno para que se puedan comunicar entre los vendedores, administradores y los súper administradores. Esto facilitaría si los vendedores tiene alguna inquietud o algún problema con algún cliente.

- Se puede llevar el proceso en el que el cliente ve cómo va su producto comprado, permitiéndoles entrar a la plataforma de una forma restringida para la integridad de los datos. Pero este proceso ya lo hacen con otro programa llamado trello, pero sería bueno integrar todo en un mismo programa.

BIBLIOGRAFIA

- Choy, K.L; Fan, K; Lo, V. (2003): “Development of an intelligent customersupplier relationship management system: application of case-based reasonning”. Industrial Management +Data System, vol.103, n°3/4, pp. 263-274
- Consejos para hacer buen responsive design [en línea], [citado en 7 de enero del 2015] < http://www.genbetadev.com/desarrollo-web/7-consejos-para-hacer-buen-responsive-design>
- FrameWorks PHP Para El Desarrollo Ágil De Aplicaciones Web [en línea], [citado en 2 de enero del 2015] < http://www.hostdime.com.co/blog/6-frameworks-php-para-el-desarrollo-agil-de-aplicaciones-web/>
- Garrido, A. (2008). La gestión de relaciones con clientes (CRM) como estrategia de negocio: Desarrollo de un modelo de éxito y análisis empírico en el sector hotelero español.pp.54
- Garrido, A. (2008). La gestión de relaciones con clientes (CRM) como estrategia de negocio: Desarrollo de un modelo de éxito y análisis empírico en el sector hotelero español.pp.54-61
- José Alberto Delgado Florentino (2007), Taller Base de datos.pp.1
- Los diferentes lenguajes de programación para la web [en línea], [citado en 19 de octubre del 2014] < http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/>
- Plakoyinnaki, E; Tzokas, N (2002): “Customer Relationship Management: A capabilities portfolio perspective”. Journal of Database Management, Vol.9,n°3,pp. 228-238.
- Google, Api Maps[en línea], [citado 22 de septiembre 2014],<http://maps.googleapis.com/maps/api/staticmap?center=Brooklyn+Bridge,New+York,NY&zoom=13&size=600x300&maptype=roadmap&markers=color:blue%7Clabel:S%7C40.702147,-74.015794&markers=color:green%7Clabel:G%7C40.711614,-74.012318&markers=color:red%7Ccolor:red%7Clabel:C%7C40.718217,-73.998284&sensor=false>
- Aniorte, teoría general de sistema[en línea], [citado en 22 octubre de 2013],<http://www.aniorte-nic.net/apunt_terap_famil_2.htm>
- Colaboración [en línea], [citado en 22 de septiembre de 2014] <http://definicion.de/colaboracion/>
- Empresamia, servidor web [en línea], [citado en 23 de septiembre de 2014] <http://empresamia.com/crear-empresa/crear/item/644-que-es-un-servidor-web>
- MySQL[en linea], [citado en 29 de septiembre de 2014] < http://www.mysql.com/>
- PHP[en linea], [citado en 29 de septiembre de 2014] <http://php.net/>
- Proyectos Agiles, Scrum[en linea], [citado en 29 de septiembre de 2014] <http://www.proyectosagiles.org/que-es-scrum>

- Enrique Gonzales, PHP[en línea], [citado 22 agosto de 2014],<http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193>
- Gestión de la información[en línea], [citado en 22 agosto de 2014], http://www.accioncontraminas.gov.co/accion/Paginas/GestionInformacion.aspx
- LARAVEL[en línea], [citado en 22 agosto de 2014], http://laravel.com/

ANEXO

MANUAL DE USUARIO

Descripción de la aplicación

La aplicación es un sistema de control de clientes, el cual permite la supervisión de los clientes de la empresa, así como los productos y la cantidad de ventas realizadas asignando un vendedor encargado a cada cliente, un administrador por zona y un súper-administrador.

A parte permite el contacto del vendedor con sus supervisores, el abordar a los clientes con promociones, y proporciona reportes para el control continuo de los productos y vendedores.

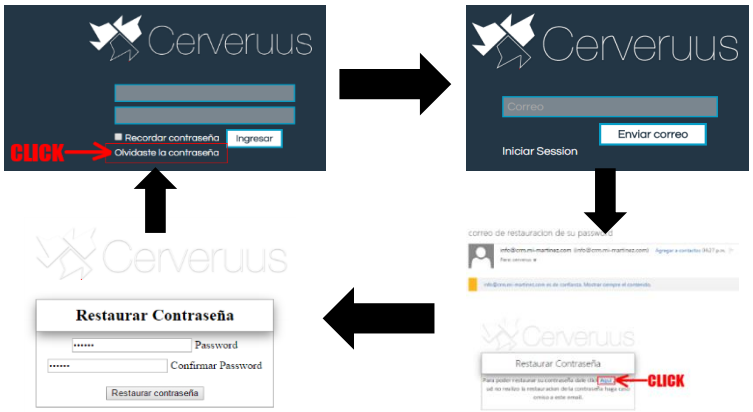
Manual de usuario

Inicio sesión

En la página principal se pedirá el correo y contraseña que serán proporcionadas por el súper-administrador o administrador para cada nuevo usuario en el sistema.

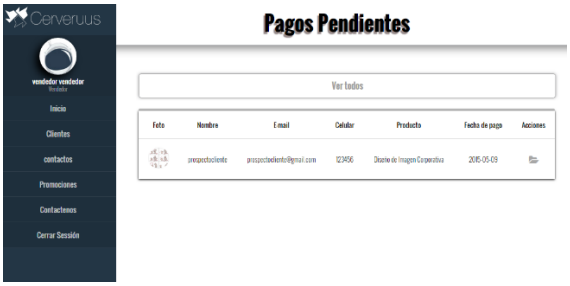


En caso de olvidar la contraseña clickea en “Olvidaste la contraseña” y en el formulario que aparece ingresa tu correo electrónico, revísalo e ingresa al link proporcionado para así definir una nueva contraseña e ingresa nuevamente al sistema.



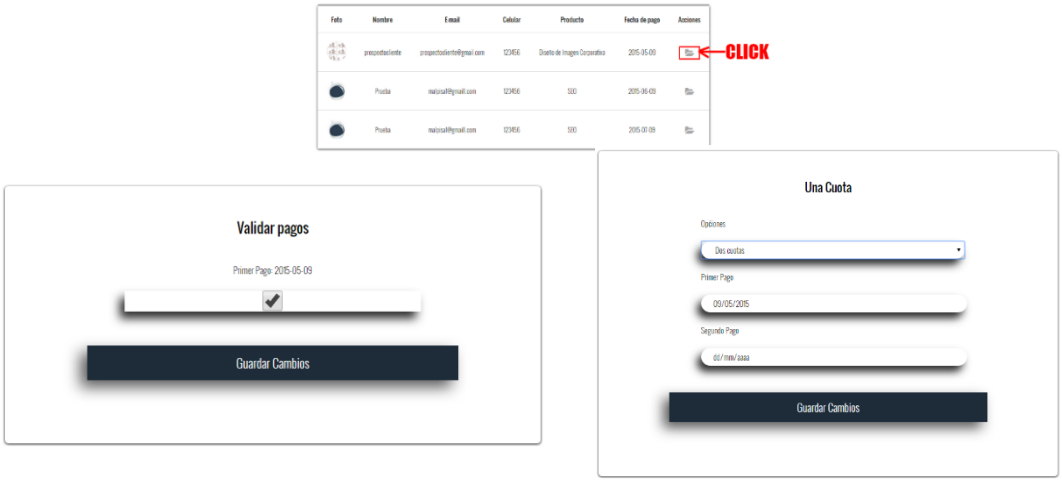
Bienvenida

Al iniciar sesión el usuario vera la cantidad de pagos pendientes de los clientes que tiene a su supervisión o al dar click en ver todos visualizara todos los pagos pendientes.





Al hacer click en la imagen de carpeta en acciones se podrá validar los pagos efectuados por el cliente o modificar la fecha de pago y sus cuotas (máximo 3).



La aplicación dispone de un menú, a continuación se describe cada ítem y su funcionamiento.

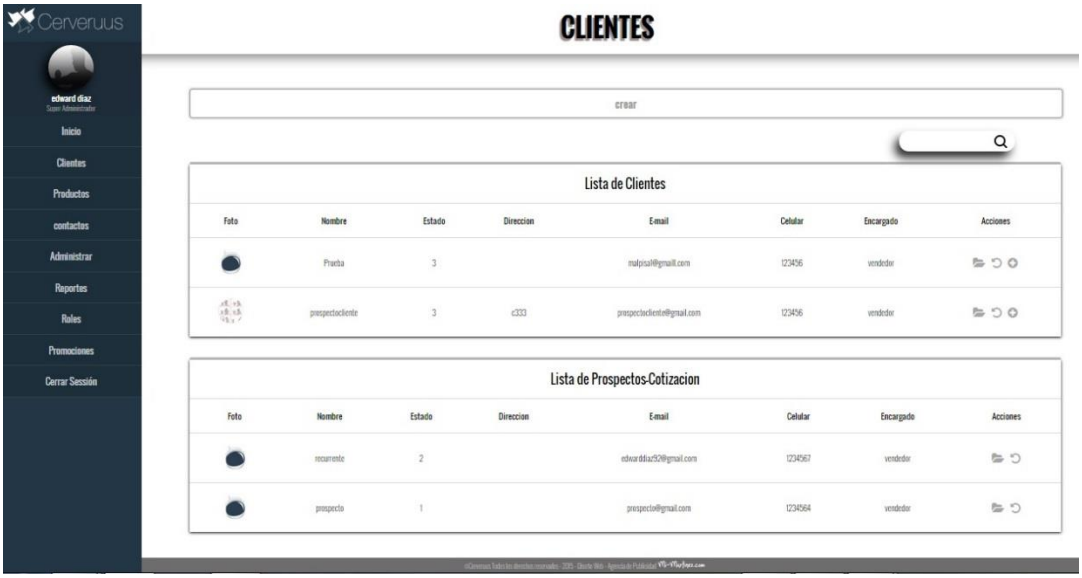
Botón de búsqueda



En la mayoría de tablas contiene este buscador según su complejidad, este permite buscar algún registro con el nombre u otros campos que contenga la tabla.

Clientes

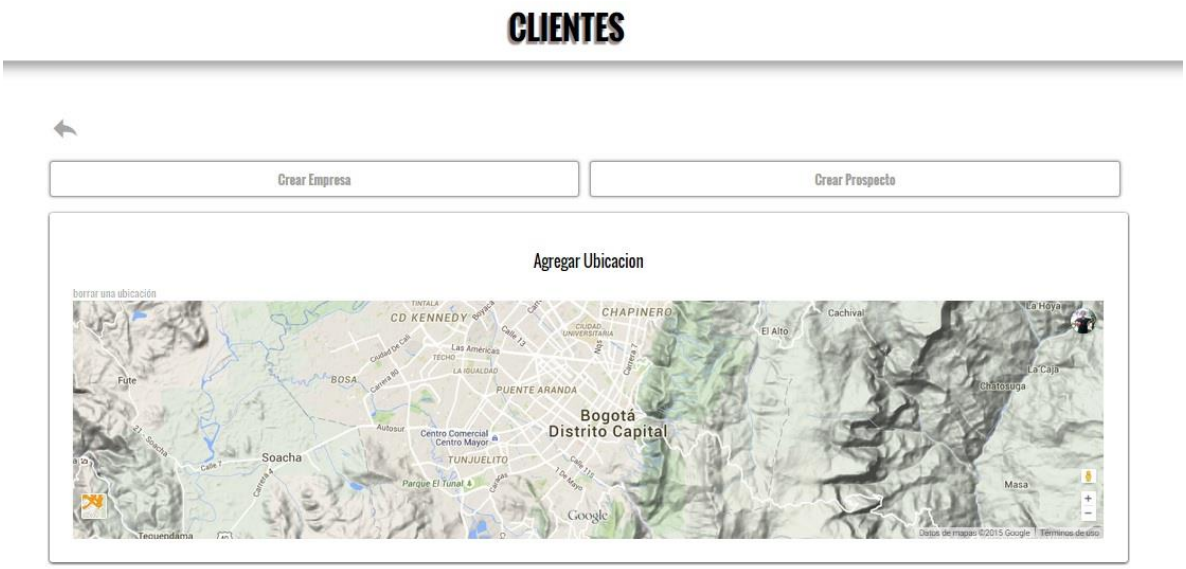
Al oprimir en clientes se visualiza los clientes de mi-martinez y los prospectos (empresas que están en proceso de ser clientes). Los clientes y prospectos tienen 2 acciones en común que sería la carpeta para ver más sobre el cliente, la flecha en círculo que es para actualizar los datos y en cliente tiene un circulo con un más en el medio que es para agregar los productos comprados por el cliente. Y al comienzo hay un botón que es para crear clientes o prospectos, además del botón de búsqueda que permite encontrar de forma muy sencilla a los clientes y prospectos



Al hacer click en crear visualizaremos el formulario de creación de clientes y prospecto.



En el formulario de creación podremos ver el botón de volver, el botón de crear empresa o crear prospecto, además tenemos un mapa para agregar la ubicación de la empresa dándole click en cualquier sitio del mapa y si nos equivocamos hay un botón que dice borrar una ubicación para volver a colocar el sitio.



Al darle click en Crear empresa visualizaremos el formulario de creación de empresas que contiene los campos de encargado, nombre, estado, nit, dirección, teléfono, extensión, correo, correo opcional, celular, pagina web, fax, país, ciudad, Skype, fuente del cliente, fecha de pago, fecha de expedición, logo y ubicación.

Crear Cliente

Encargado

seleccione un encargado

Nombre

Estado

seleccione un estado

Nit

Dirección

Telefono

Extensión

Correo electronico

Correo electronico opcional

Celular

Pagina Web

Fax

Pais

Ciudad

Skype

Fuente del Cliente

Fecha de pago

dd/mm/aaaa

Fecha de pago

dd/mm/aaaa

Fecha de Expedicion

dd/mm/aaaa

Logo

Seleccionar archivo Ningún archivo seleccionado

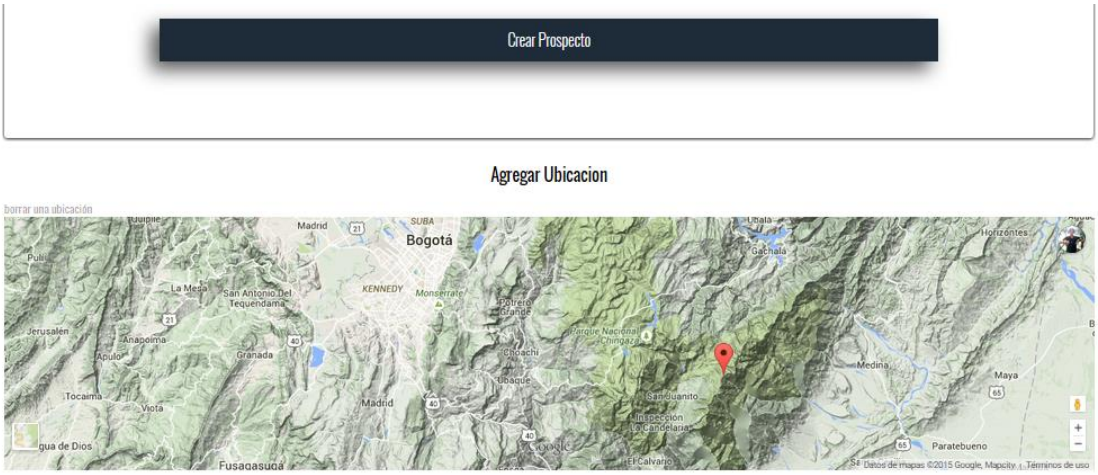
Crear Cliente

Agregar Ubicacion



Al darle click en Crear prospecto visualizaremos el formulario de creación de empresas que contiene los campos de encargado, nombre, estado, nit, dirección, teléfono, extensión, correo, correo opcional, celular, pagina web, fax, país, ciudad, Skype, fuente del cliente, logo y ubicación.





Al darle click en el botón volver.



Volvemos al inicio de clientes



Al darle click en la acción de la carpeta visualizaremos los datos del cliente y demás.

Infomarción del Cliente



Nombre: Prueba

Encargado: vendedor

Estado: Moroso

Nit: 123456

Direccion:

Telefono: 123456

Ext: 0

E-mail: malpisal@gmaill.com

E-mail Opcional: 0

Celular: 123456

Pagina web:

Fax: 0

Pais:

Ciudad: Bogota

Skype:

Fuente:

Fecha de pago: 2015-05-19

Fecha de expedición: 2015-07-20

agregar un producto a cotizar

Estado del cliente

- cotizacion oridnaria
- Portafolio y propuesta
- Primera llamada
- Cotizacion especifica
- Segunda llamada
- Negociacion
- Activo
- Mensual
- Moroso
- Eliminado

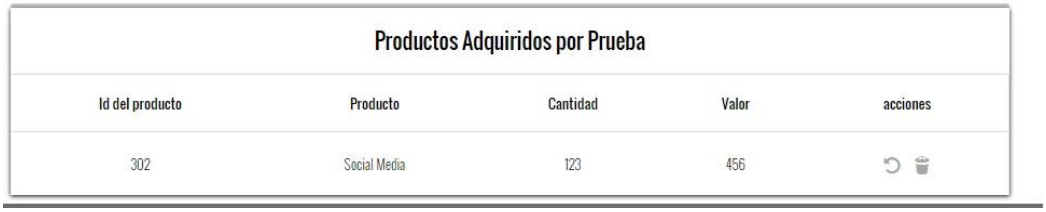
Guardar Cambios

Que tiene un botón de crear contactos directos con la empresa, el botón de agregar un producto a cotizar, para hacer un historial de que producto se está cotizando y en el estado del cliente podremos visualizar los estados que son un menú desplegable que muestra el contenido de ese estado. Al dar click en el botón de contactos se despliega un menú de creación de contactos.

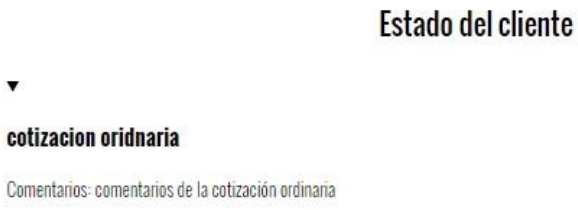


Al dar crear contacto se enviara al formulario de creación de contactos, pero este módulo lo explicaremos más adelante. Al darle click en agregar un producto a cotizar se muestra los productos que ha cotizado y la posibilidad de crear, actualizar o eliminar.





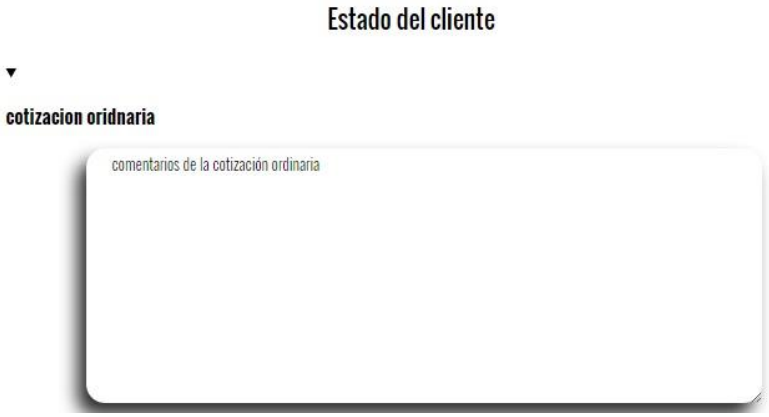
Podremos visualizar un formulario donde se crear los productos a cotizar, si se quiere actualizar se oprime en la tabla en donde está el icono de la flecha en círculo y los datos pasaran al formulario para que sean editados o si quieren eliminar el producto cotizado solo hay que darle click en el icono de caneca de basura. Ahora visualizaremos los estados del cliente.



Podemos darnos cuenta que podemos desplegar la cotización ordinaria que contendrá los comentarios de ese estado. Si se procede a editar o agregar más comentarios solo le damos click en comentarios.



Se podrá modificar o agregar más líneas al comentario.



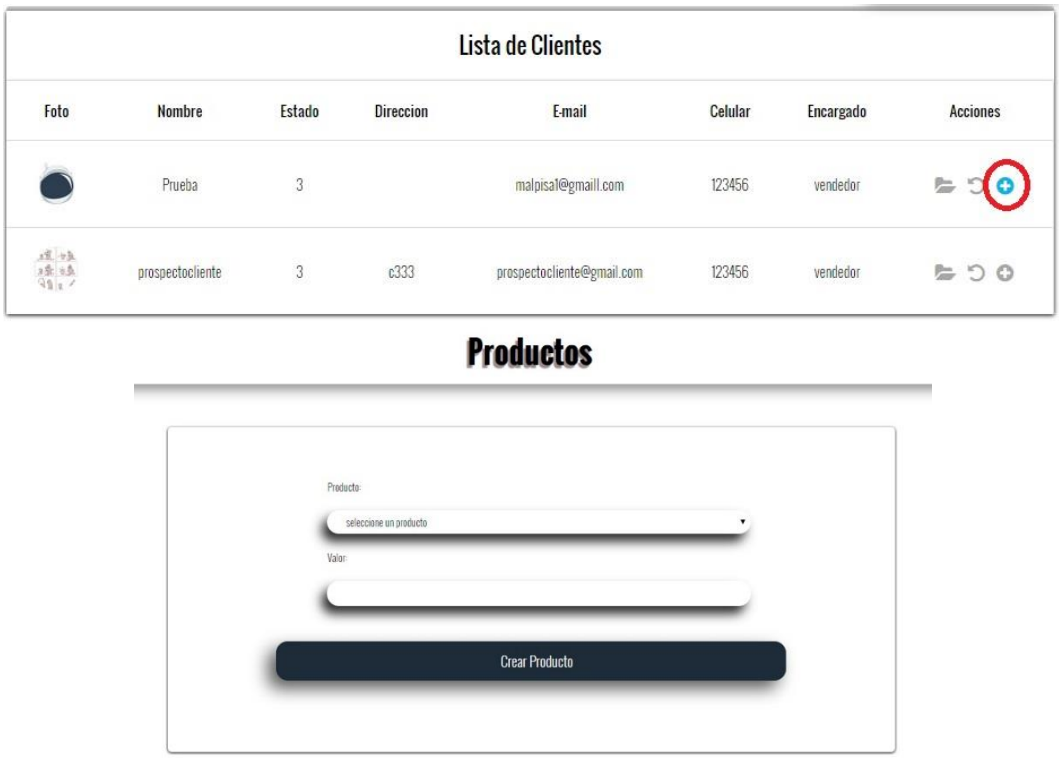
Y dándole click al botón guardar cambios salvaremos los comentarios hechos por el usuario.



En el módulo de clientes al darle click en actualizar se despliega el formulario de creación con los datos de la empresa para ser modificados.



Ahora cuando le damos click en el icono de agregar productos comprados nos enviara a un formulario de creación de productos adquiridos por el cliente y una tabla visualizando los productos comprados y sus acciones.





Aquí podremos crear productos en el formulario o eliminarlos. Pero al dar click en la carpeta podremos modificar las fechas y las cuotas a la que se deben pagar el producto.



Aquí podemos ver el formulario para ver a cuantas cuotas queda el producto y que fechas. En opciones podemos dar la cantidad de cuotas, que sería de una a tres, y dependiendo de cuál se escoja se despliega la cantidad de fechas requeridas para los pagos y lo guardamos con el botón guardar cambios.



Al guardar los cambios en la misma página nos aparece las fechas guardadas anteriormente y con un capo de validación para verificar si los pagos fueron realizados.



Al hacer las validaciones podemos guardar los cambios en el botón de guardar cambios.

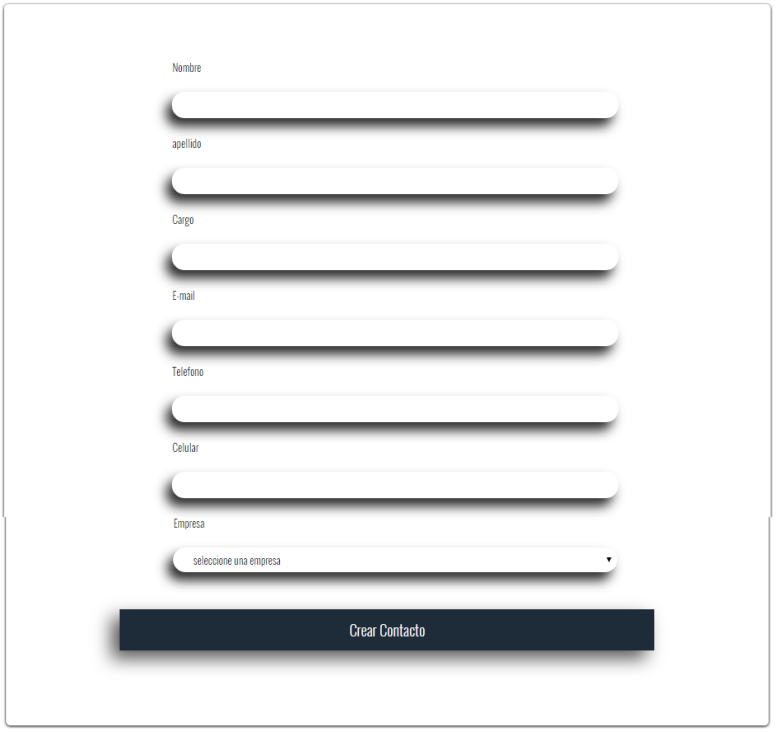


Contactos

En este módulo podemos observar que tenemos una lista de contactos, estos contactos están relacionados a una empresa, por ejemplo un gerente o secretaria y en este módulo podemos crear o actualizar los contactos. Al darle click en actualizar se muestra un formulario con los datos del contacto para que puedan ser actualizados.

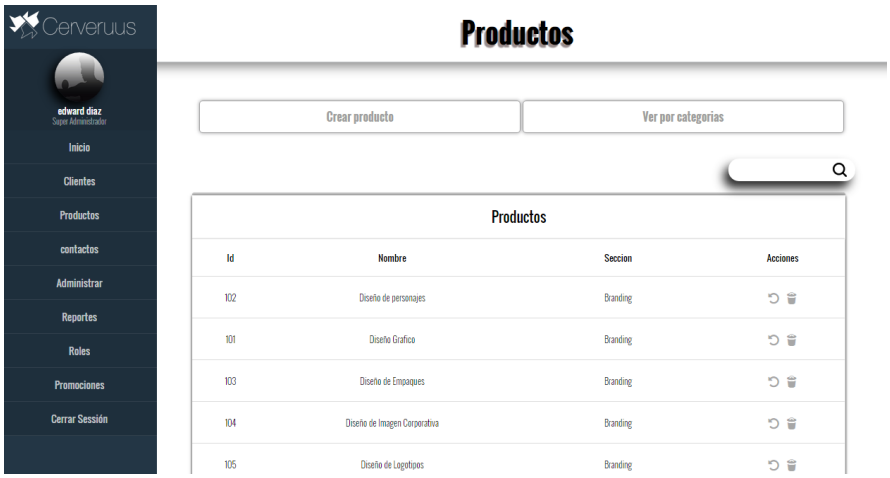


Al darle click en nuevo contacto se visualiza el formulario de creación de contactos.

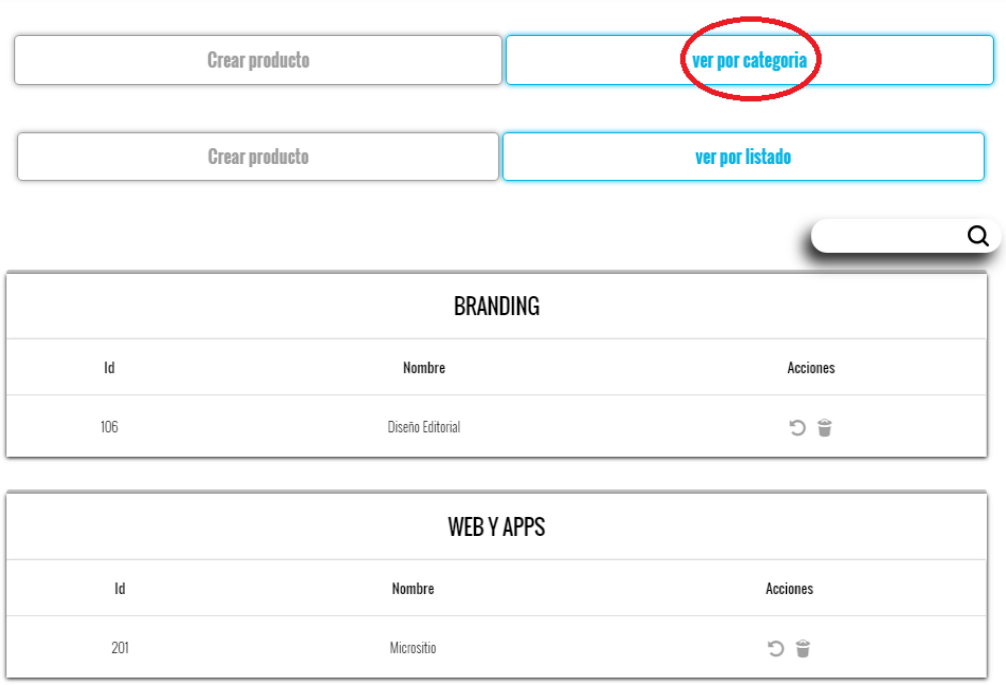


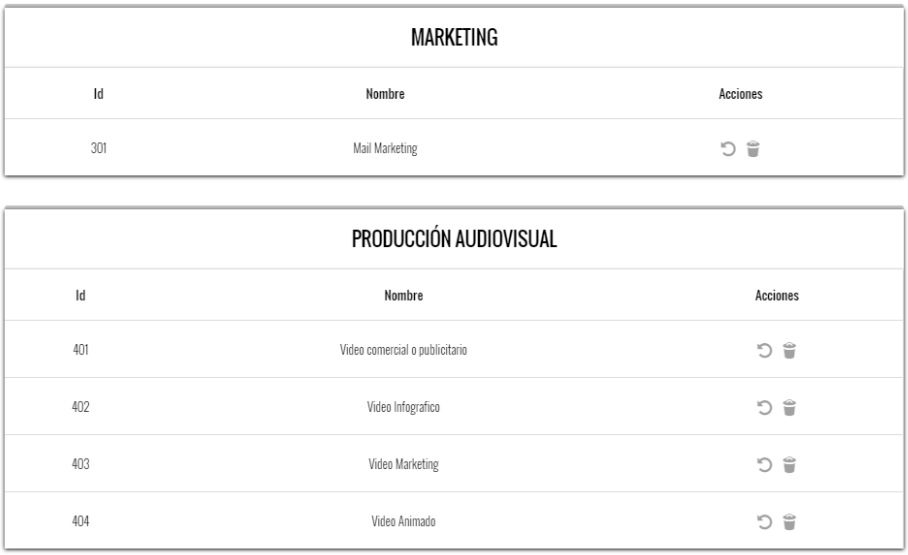
Productos

El módulo de productos es para ver, editar o eliminar productos que tiene a servicio la empresa mi-martinez, los productos están en una lista normal, pero podemos verla también por categorías

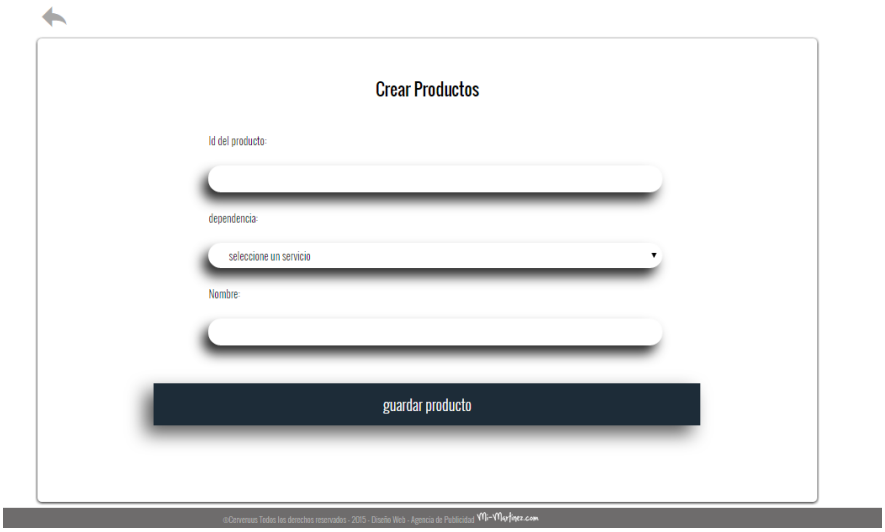


Al darle click en categorías podremos visualizar los productos por categorías y el botón de categorías cambia su texto a “ver por listado” y si le damos click otra vez vuelve a su estado anterior



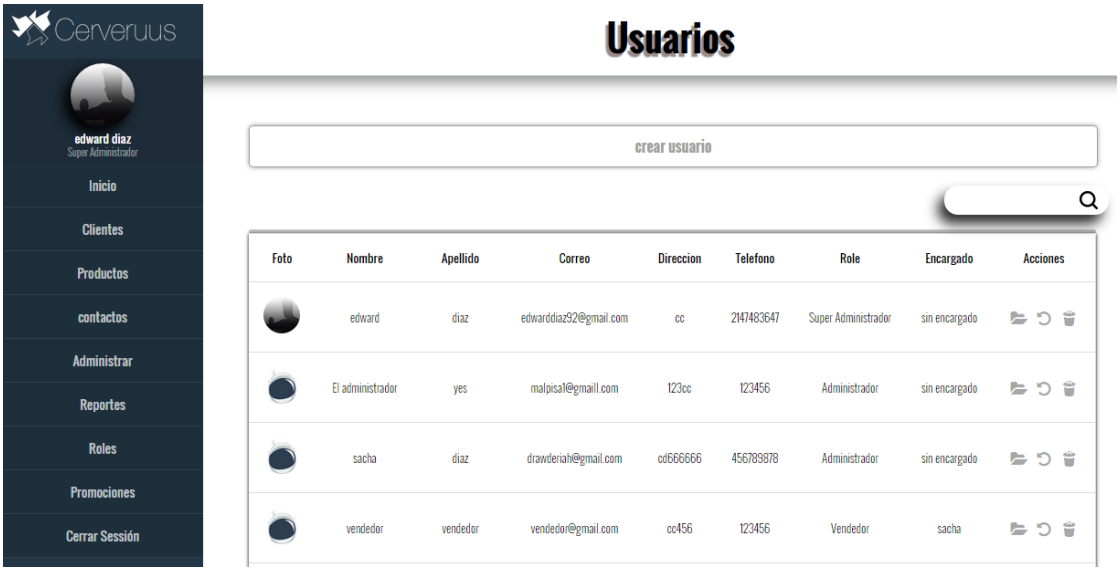


Al darle click en crear producto nos redirige al formulario de creación de productos que contiene el botón de volver y el formulario de creación donde se llenan todos los campos y se da click en el botón de guardar producto.



Administrar

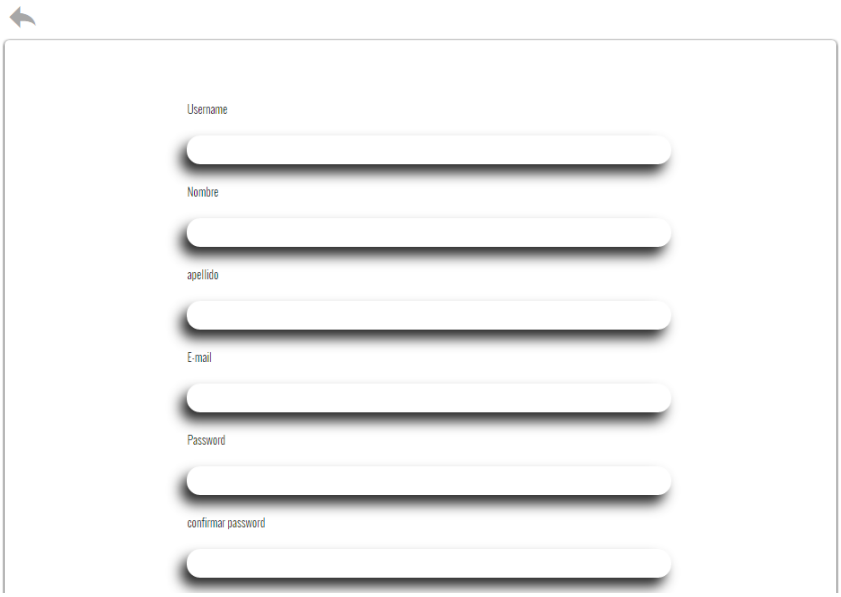
Aquí podemos ver todo lo relacionado con los usuario que tienen acceso a la plataforma, en este módulo se filtra la información según el tipo de cuenta que posea, si eres súper administrador podemos ver todos los usuarios, si eres administrador solo puedes ver los vendedores a su cargo y si eres vendedor no podrás ver nada, aunque el vendedor no puede acceder al módulo de administrar en este caso veremos la cuenta súper administrador.

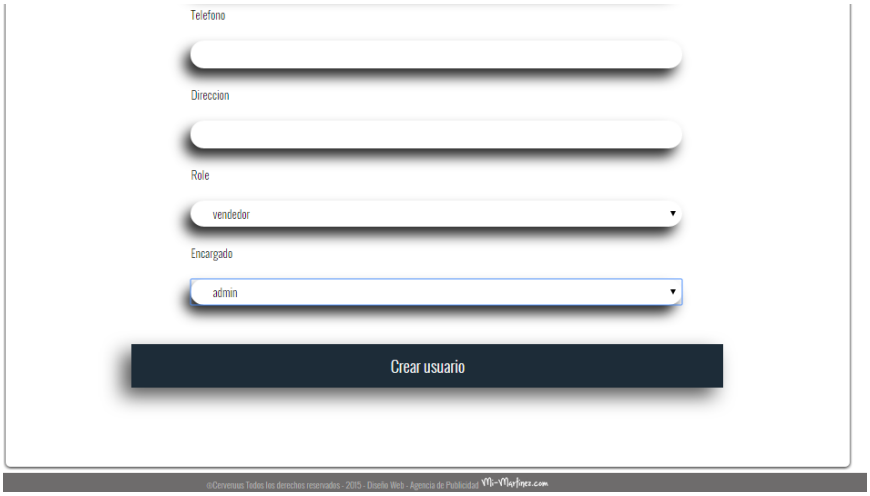


Al darle click en crear usuario nos enviara al formulario de creación de usuarios, esta vista si eres super administrador puedes crear cualquier tipo de cuenta, si eres administrador solo puedes crear el tipo de cuenta vendedores y los vendedores no pueden crear cuentas.



Esta vista contiene el botón de volver y el formulario de creación que tiene unos campos que son obligatorios para su creación como: username, nombre, apellido, email, password, teléfono, dirección, role y encargado si el tipo de cuenta es vendedor.





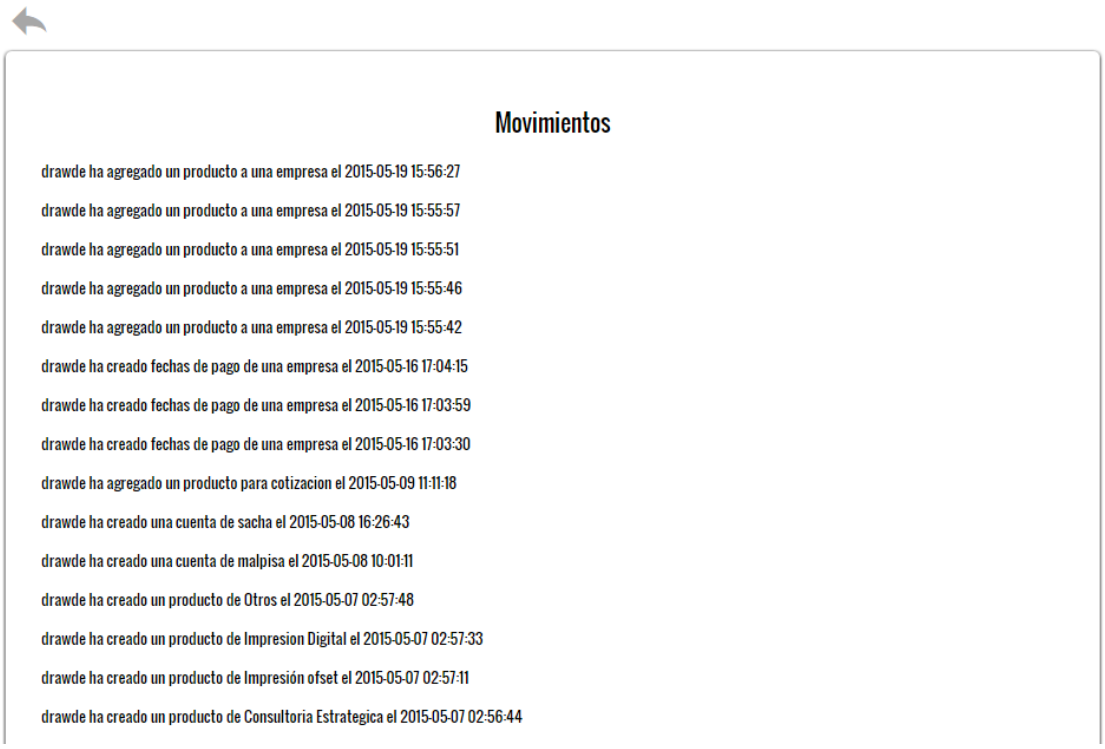
En la tabla de usuarios también podemos ver que tiene 3 acciones ver, actualizar y eliminar.



Al darle click en actualizar (icono de la flecha en círculo) podemos ver los datos en el formulario de creación para que puedan ser modificados, al darle click en eliminar (icono de la caneca de basura) podemos eliminar el usuario y en ver (icono de la carpeta) podemos ver más información del usuario, como los movimientos dentro de la página, si eres administrador además aparecerá los vendedores que tienes a cargo o si eres vendedor saldrán los clientes a cargo. Ahora vamos a darle click en la carpeta.



En el súper administrador aparecerán los movimientos y el botón de volver.



En el administrador aparecerán los usuarios que tiene a cargo, los movimientos y el botón de volver.



En el vendedor aparecerán los clientes y prospectos que tiene a cargo, los movimientos y el botón de volver.



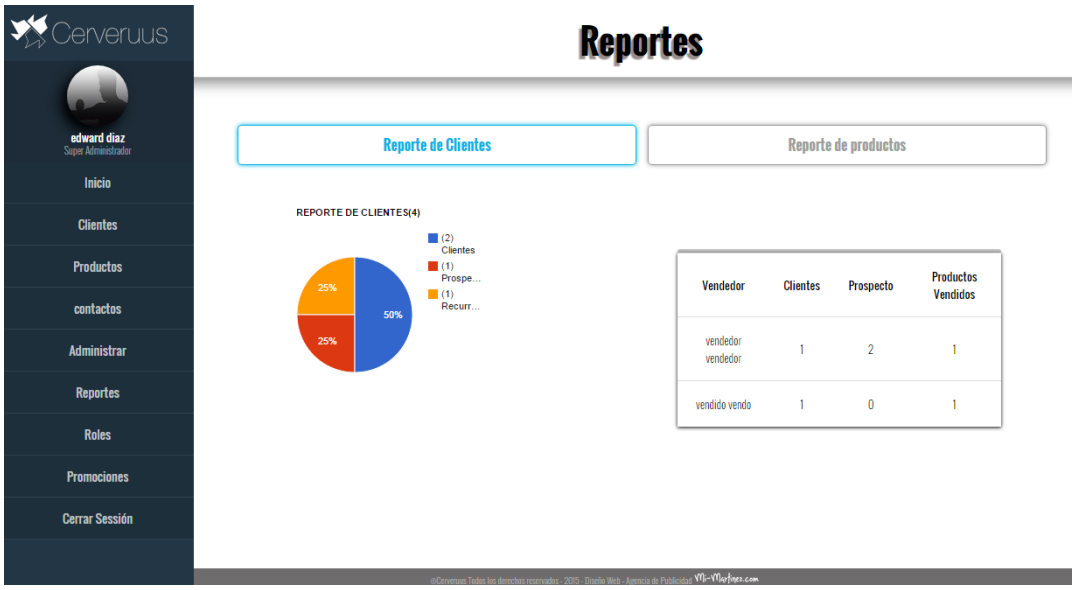
Al darle click en movimientos se despliegan los movimientos realizados por esa cuenta.



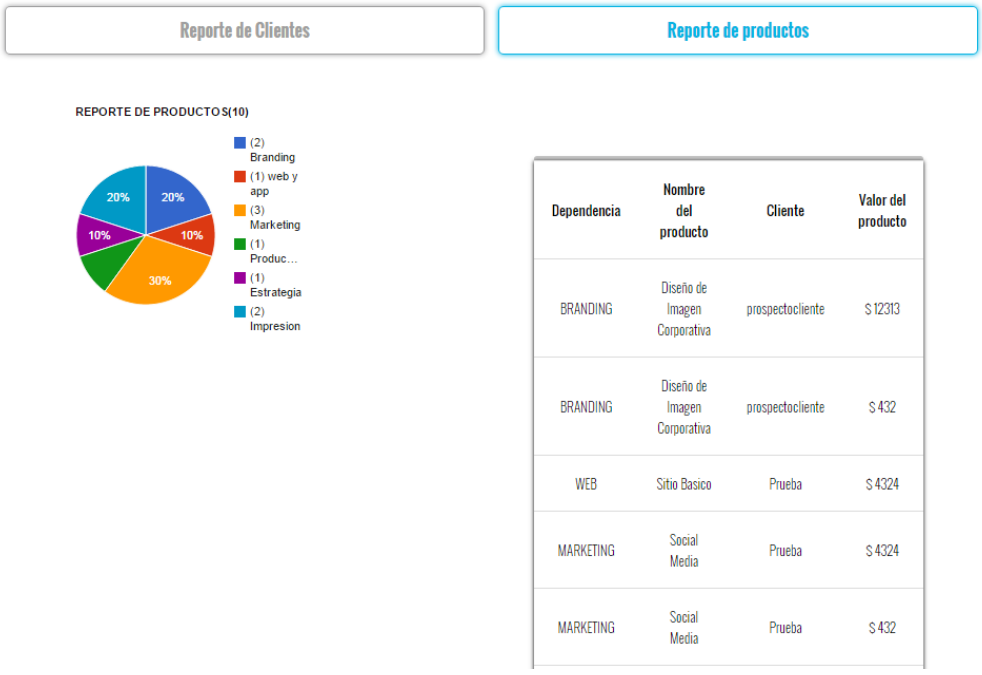


Reportes

En este módulo tenemos los reportes de clientes y reporte de productos. Cuando vemos los reportes aparecerá primero el reporte de clientes.

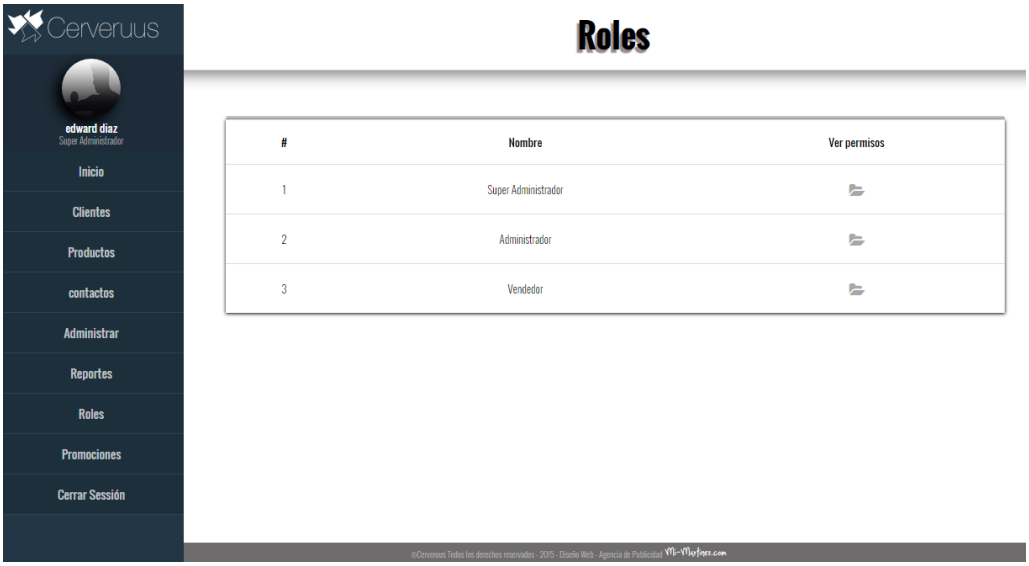


Podemos visualizar una gráfica y una tabla con los datos de los clientes y al darle click en reporte de productos podemos visualizar lo mismo pero con los datos de los productos.

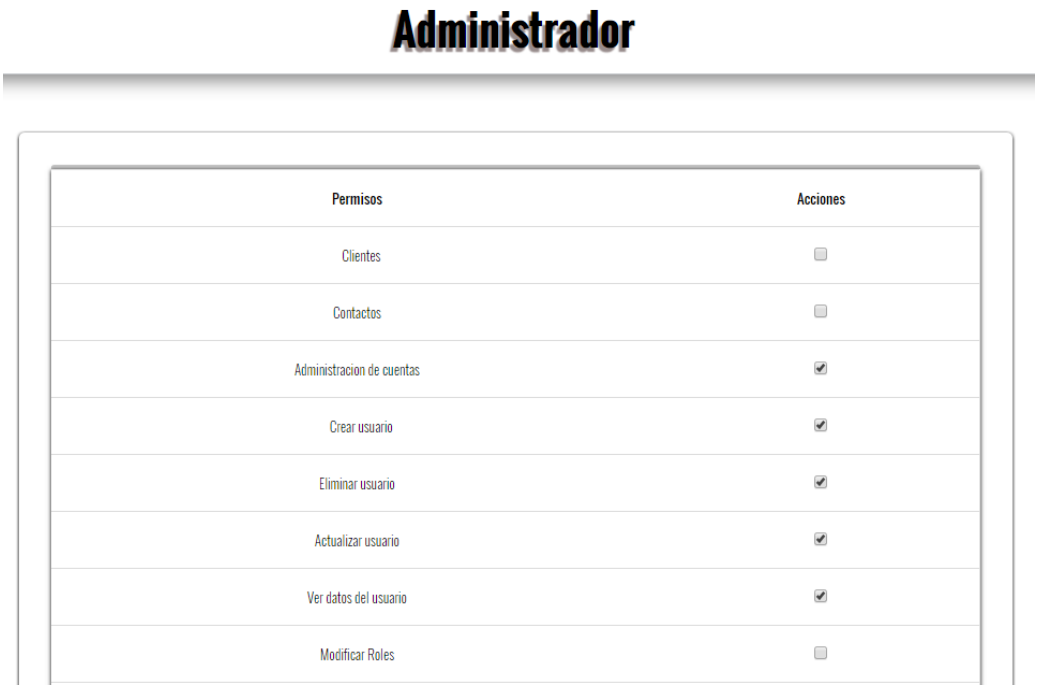


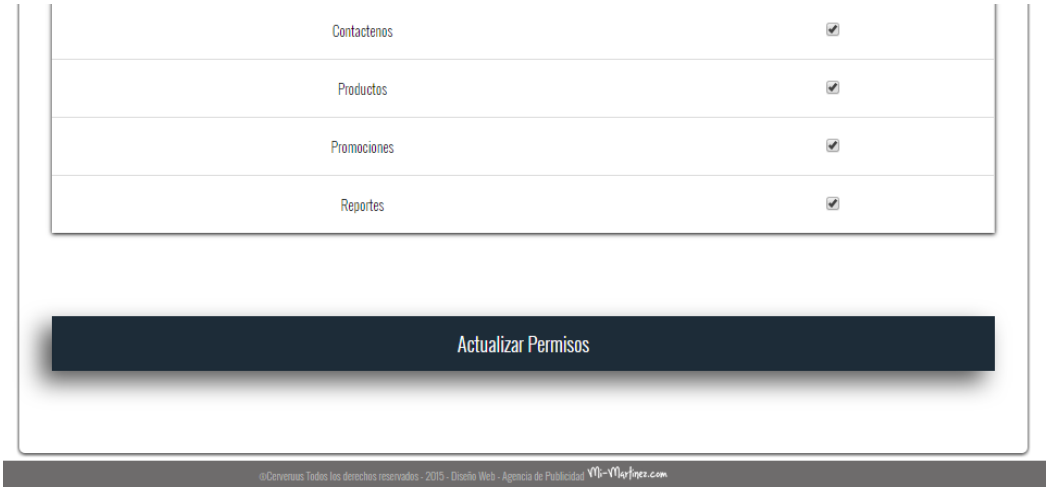
Roles

En este módulo podemos visualizar una tabla con los tipos de cuenta que tiene la plataforma con una acción que es modificar los permisos que tiene cada tipo de cuenta. En el caso del súper administrador no se pueden cambiar los permisos, solo se pueden visualizar.



Al darle click en la carpeta podemos visualizar una especie de formulario donde podemos modificar los permisos.





En esta vista podemos ver que permisos tiene este tipo de cuenta y podemos modificar los permisos y darle click en actualizar permisos.

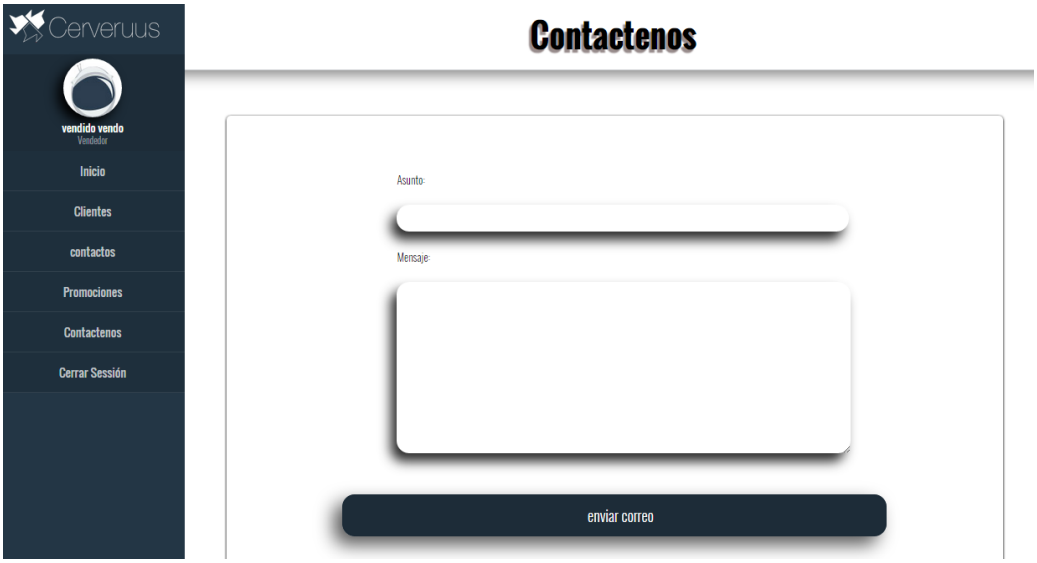
Promociones

En este módulo podemos ver un formulario donde llenamos los campos y envía correos a todos los clientes, este módulo fue creado para enviar información a los clientes y prospectos de los servicios de mi-martinez.com



Contáctenos

En este módulo podemos enviar un correo si tenemos un inconveniente o alguna duda con relación a la plataforma o los clientes.



Perfil

En este módulo podemos ver la información de nuestra cuenta además puedes cambiar la contraseña, actualizar los datos y ver sus movimientos. Al darle click en la foto de su cuenta podrás acceder al perfil.

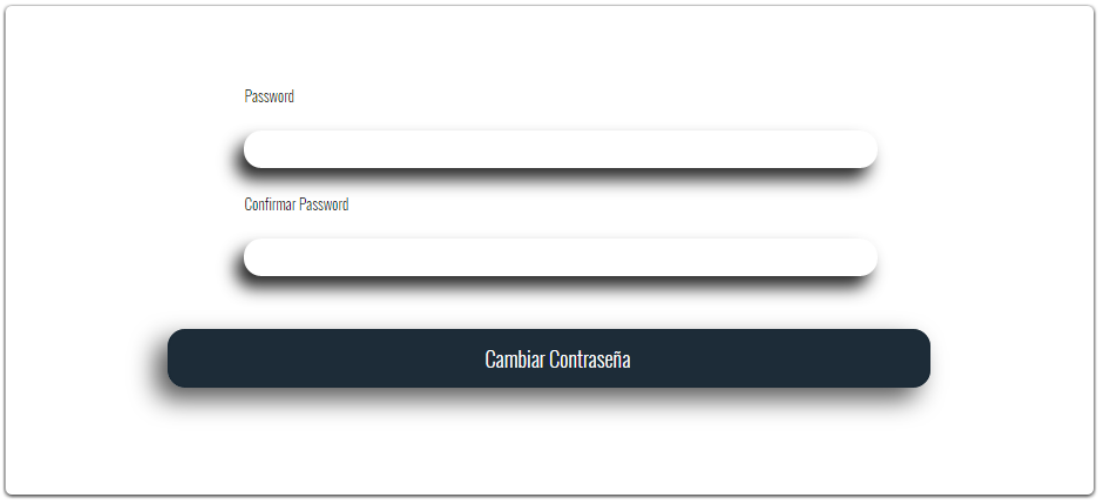




Al darle click en movimientos ya sabemos que se despliega los movimientos hechos por usted en la plataforma. Al darle click en acutalizar datos nos envía al formulario donde se podrá cambiar los datos personales y es el mismo formulario de creacion de usuarios.



Al darle click en cambiar contraseña se visualizara un formulario en el que podrás cambiar tu contraseña



En esta vista solo hay que llenar los campos y deben ser iguales, luego se da click en cambiar contraseña y se guardara la nueva contraseña.